行政院國家科學委員會專題研究計畫 成果報告

多目標演化式演算法理論基礎之探討：不均衡性，群族大小和收斂時間(第 2 年)
研究成果報告(完整版)

計 畫 類 別 ： 個別型
計 畫 編 號 ： NSC 96-2221-E-216-037-MY2
執 行 期 間 ： 97 年 08 月 01 日至 98 年 07 月 31 日
執 行 單 位 ： 中華大學資訊工程學系

計 畫 主 持 人 ： 陳建宏

計畫參與人員： 碩士班研究生-兼任助理人員：李誌堅
碩士班研究生-兼任助理人員：魏良哲
碩士班研究生-兼任助理人員：康志瑋

報 告 附 件 ： 出席國際會議研究心得報告及發表論文

處 理 方 式 ： 本計畫涉及專利或其他智慧財產權，2 年後可公開查詢

中 華 民 國 98 年 11 月 03 日

行政院國家科學委員會補助專題研究計畫 ■ 成 果 報 告 □期中進度報告

多目標演化式演算法理論基礎之探討: 不均衡性, 群族大小和收斂時間

計畫類別：■ 個別型計畫　　□ 整合型計畫
計畫編號：NSC　－　　－　－　　－　　－
執行期間：　96 年　8　月　1　日至　98　年　7　月　31　日

計畫主持人：　陳建宏
共同主持人：
計畫參與人員：　康志瑋、陳建宏、盧金榮、魏良哲、李誌堅

成果報告類型(依經費核定清單規定繳交)：□精簡報告　■完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
■出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年■二年後可公開查詢

執行單位：中華大學資訊工程學系

中　華　民　國　　98　年　10　月　31　日

I、中文摘要

許多現實生活上的最佳化問題常需要最佳化多個不同尺度且互相衝突競爭的目標，這些問題通常被稱為多目標最佳化問題。相對於單目標最佳化問題只需求解一個單一的最佳解答，多目標最佳化問題最大的差異處在於其必須求解出許多個最佳解答。近年來，由於多目標演化式演算法可以在一次的執行中有效地同時搜尋多個解答，因此多目標演化式演算法被廣泛認為非常適用於求解多目標最佳化問題。儘管多目標演化式演算法已被廣泛認為適用於求解現實生活中的許多多目標最佳化問題，絕大多數的研究卻僅限於針對該研究領域的問題來設計有效的多目標演化式演算法。僅僅有非常少數的研究探討多目標演化式演算法在求解多目標最佳化問題時所需的族群大小和收斂時間。

本研究計畫的主題在於探討多目標演化式演算法在解決具有各種不均衡特性的多目標最佳化問題之效能。本計劃之研究結果除了在學理上可以提供改良多目標演算法之貢獻外，並可以提供工業界在應用多目標最佳化演算法之參數決定問題。本計劃之研究成果並已部份實踐於數篇會議論文之中，並加以延伸投稿期刊論文。

關鍵字：多目標最佳化、演化式演算法

II、Abstract

Many real-world optimization problems involve multiple incommensurable and often competent objectives; these problems are known as multi-objective optimization problems (MOOPs). Many MOOPs cannot satisfactorily be characterized by a single performance measure. Due to the nature of trade-offs involved, MOOPs seldom have a unique solution. Instead of obtaining a single optimal solution, the ultimate goal of solving MOOPs is to find a complete set of Pareto-optimal solutions. Recently, multi-objective evolutionary algorithms (MOEAs) have been recognized to be well-suited for solving MOOPs because their abilities to exploit and explore multiple solutions in parallel and to find a widespread set of non-dominated solutions in a single run. Although MOEAs have been shown to be effective for solving many real-world applications and exploring complex non-linear search spaces as efficient optimizers, but only a few preliminary analysis based on selectorecombinative MOEAs and (1+1)MOEA have been conducted in analyzing the population sizing and convergence time of MOEAs in solving MOOPs.

The main topics of this project are to investigate the performances of MOEAs in solving MOOPs with disequilibrium, and study the important factors that affect the convergence time and population sizing of MOEAs. These models can provide practitioners guidance in choosing key MOEAs parameters, and also assists MOEA practitioners to get maximum mileage on designing their MOEAs. The results of this project have been published in several conference papers, and their extended results have been submitted for the review of journals.

III、**Background, Motivation, and Objectives**

Multi-objective optimization problems (MOOPs) are common in our real life. A MOOP has a number of objective functions to be maximized or minimized. For example, consider the design of a car. Generally, the cost of such systems is to be minimized, while maximum performance is desired. Depending on conditions of the application, further objectives may be important such as reliability and energy dissipation. Considering the design of a car, and assuming that the two objectives cheapness (f_1) and performance (f_2) are to be maximized under speed constraints. Then, an optimal design might be an architecture which achieves maximum performance at minimal cost and does not violate the speed constraint. However, what makes MOOPs difficult is that a solution may be optimal in an objective function, but bad in other objective functions. The objectives are conflicting and cannot be optimized simultaneously. Instead, a satisfactory trade-off has to be found. In the example of designing a car, cheapness (the inverse of cost) and performance are generally competing. High-performance car architectures substantially increase costs, while car architectures with cheap costs usually provide low performance. Depending on the market requirements, an intermediate solution (medium performance, medium cost) may be an appropriate trade-off for decision makers.

There are many industrial applications belong to MOOPs. Take a manufacturing factory for another example, production planning have to consider routing optimization, equipment optimization and machine optimization. Take an IC design application for an example, in the layout processes of an IC, the floorplan process usually seeks to optimize two competing objectives: area and routeability; and the result of the placement and routing depend on the result of floorplanning.

Assume the multi-objective functions are to be minimized. Mathematically, MOOPs can be represented as the following vector mathematical programming problems:

$$Minimize\ \ F(Y) = \{f_1(Y), f_2(Y), ..., f_i(Y)\}. \tag{1}$$

where Y denotes a solution and $fi(Y)$ is generally a nonlinear objective function. Pareto dominance relationship and some related terminologies are introduced below. When the following inequalities hold between two solutions Y_1 and Y_2, Y_2 is a non-dominated solution and is said to dominate Y_1 ($Y_2 \succ Y_1$):

$$\forall i\ :\ f_i(Y_1) > f_i(Y_2)\ \wedge \exists j\ :\ f_j(Y_1) > f_j(Y_2). \tag{2}$$

When the following inequality hold between two solutions Y_1 and Y_2, Y_2 is said to weakly dominate Y_1 ($Y_2 \succeq Y_1$):

$$\forall i\ :\ f_i(Y_1) \geq f_i(Y_2)\ . \tag{3}$$

A feasible solution Y * is said to be a Pareto-optimal solution if and only if there does not exist a feasible solution Y where Y dominates Y *, and the corresponding vector of Pareto-optimal solutions is called Pareto-optimal front.

The great success for evolutionary computation techniques, including evolutionary programming (EP), evolutionary strategy (ES), genetic algorithm (GA), came in the 1980s when extremely complex optimization problems from various disciplines were solved, thus facilitating the undeniable breakthrough of evolutionary computation as a problem-solving methodology. Inspired from the mechanisms of natural evolution, evolutionary algorithms (EAs) utilize a collective learning process of a population of individuals. Descendants of individuals are generated using randomized operations such as mutation and recombination. Mutation corresponds to an erroneous self-replication of individuals, while recombination exchanges information between two or more existing individuals. According to a fitness measure, the selection process favors better individuals to reproduce more often than those that are relatively worse. Specifically, GAs are used to illustrate the basic framework of EAs. GAs are stochastic, population-based search and optimization algorithms loosely modeled after the paradigms of evolution. GAs guide the search through the solution space by using natural selection and genetic operators, such as crossover, mutation, and the like. EAs have been shown to be effective for solving NP-hard problems and exploring complex non-linear search spaces as efficient optimizers. The robust capability of EAs to find solutions to difficult problems has permitted them to become a popular optimization and search technique in many industries.

Recently, multi-objective evolutionary algorithms (MOEAs) have been recognized to be well-suited for solving MOOPs because their abilities to exploit and explore multiple solutions in parallel and to find a widespread set of non-dominated solutions in a single run. Several MOEAs based on Pareto dominance relationship are proposed to solve MOOPs directly, and present more promising results than single-objective optimization techniques theoretically and empirically. By making use of Pareto dominance relationship, MOEAs are capable of performing fitness assignment without using a weighted linear combination of all objectives.

The objectives of this project are to study the four important factors that affect the performance of MOEAs and to discover the relationship of these factors with convergence time and population sizing of MOEAs. By making uses of our results, we can further develop efficient multi-objective evolutionary algorithms to solve real-world application more quickly and reliable.

IV丶Results

The results of this project have been submitted for possible publication of a journal and published in the following conference papers:

[1] J-H. Chen, **Jian-Hung Chen**, “Multi-objective Memetic Approach for Flexible Process Sequencing Problems.” in *Proceeding of 2008 ACM SIG-EVO Genetic and Evolutionary Computation Conference (GECCO-2008)*, pp. 2123-2128. **(EI)**

[2] **Jian-Hung Chen**, “Memetic Approach for Multi-objective Flexible Process Sequencing Problems.” in *Proceeding of 2008 WORLDCOMP Conference (WORLDCOMP-2008),* pp. 248-254.

[3] C-W. Kang, **Jian-Hung Chen**, “Multi-objective Evolutionary Optimization of 3D Differentiated Sensor Network Deployment.” in *Proceeding of 2009 ACM SIG-EVO Genetic and Evolutionary Computation Conference (GECCO-2009)*, pp. 2059-2064. **(EI)**

[4] C-W. Kang, **Jian-Hung Chen**, “An Evolutionary Approach of Multi-Objective 3D Differentiated Sensor Network Deployment.” in *Proceeding of 12th IEEE International Conference on Computational Science and Engineering (CSE-09))*, pp. 187-193. **(EI)**

[5] C-H. Chen, **Jian-Hung Chen**, “A Multi-Objective Evolutionary Approach forCombined Heat and Power Environmental/Economic Power Dispatch” in *Proceeding of 2009 WORLDCOMP Conference (WORLDCOMP-2009).*

[6] L.-C. Wei, C-W. Kang, **Jian-Hung Chen**, “A Force-Driven Evolutionary Approach Optimization of 3D Differentiated Sensor Network Deployment.” in *Proceeding of 2009 IEEE MASS Conference (MASS-2009)* **(EI)**

Multi-objective Memetic Approach for Flexible Process Sequencing Problems

Jian-Hong Chen
Department of Computer Science and Information Engineering
707 Sec. 2 Wu-Fu Road
Hsin-Chu 300, Taiwan

Jian-Hung Chen
Department of Computer Science and Information Engineering
707 Sec. 2 Wu-Fu Road
Hsin-Chu 300, Taiwan
jh.chen@ieee.org

ABSTRACT

This paper describes a multi-objective memetic approach for solving multi-objective flexible process sequencing problems in flexible manufacturing systems (FMSs). FMS can be described as an integrated manufacturing system consisting of machines, computers, robots, tools, and automated guided vehicles (AGVs).FMSs usually pose complex problems on process sequencing of operations among multiple parts. An efficient multi-objective memetic algorithm with fitness inheritance mechanism is proposed to solve flexible process problems (FPSs) with the consideration the machining time of operations and machine workload load balancing. The experimental results demonstrate that our approach can efficiently solve FPSs and fitness inheritance can speed up the convergence speed of the proposed algorithm in solving FPSs.

Categories and Subject Descriptors

J.6 [**COMPUTER-AIDED ENGINEERING**]: Computer-aided manufacturing (CAM)

General Terms

Algorithms, Design, Performance

Keywords

process planning, flexible manufacturing systems, multi-objective optimization, memetic algorithms, fitness inheritance

1. INTRODUCTION

Computer-aided process planning (CAPP) is an automated system for preparation of a plan that specifies machines, machine conditions, operations, operation sequence, and tools required to production these components. Traditionally, the process sequencing has been solved by either the experience of process planners or a fixed and static process plan consisting of an ordered sequence of operations. However, the traditional mythologies are not suitable in real flexible environment, because the techniques have a few constraints in order to cope with dynamic situations of the flexible environment [7]. Moreover, as the number of operations increase, it poses more difficulties for decision makers to plan a cost-effective process sequences for manufacturing.

In this paper, a memetic algorithm using fitness inheritance (MEFI) is proposed to solve multi-objective flexible process sequencing problems (FPSs) having three objectives: minimizing total machining time, maximum machine workload and machine workload unbalance. The proposed approach can obtain a set of non-dominated solutions for decision makers in a single run, without the necessary of problem decomposition and relative preferences. Decision makers can easily distinguish between the costs of different process sequences and choose more than one satisfactory process sequences at a time. Six benchmark problems with different complexities are used to evaluate the performance of the proposed approach. A multi-objective genetic algorithm (MOGA) without local search and fitness inheritance is used for performance comparisons. It is shown empirically that MAFI outperforms MOGA in terms of the solution quality.

This paper is organized as follows: Section 2 presents the background of process sequencing problems, multi-objective evolutionary optimization. Section 3 introduces the setup of flexible manufacturing system and the mathematical formulation of FPSs. Section 4 presents the multi-objective memetic algorithm for solving FPSs. Section 5 presents the experimental analysis of the proposed algorithm, and Section 6 summarizes our conclusions.

2. BACKGROUND

2.1 Process Sequencing Problems

Flexible process sequencing problems are well known among the combinatorial optimization problems. Previous research focused on two important key issues of process sequencing problems, described as follows. The first key issue is the objective functions of process sequencing. Several approaches [4, 1] are proposed for process sequencing with various objectives. Another key issue that arises recently is the alternative process sequences. In the view of real time scheduling, alternative process sequences provide additional capability for the decision maker (DM) to cope with unpredictable events such as machine failures or rush orders. From the view of off-line scheduling, alternative process sequences may be used to improve the schedule quality by reducing

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-131-6/08/07.

the load on bottleneck machines [1]. It is essential but also a challenge for DM to prepare a set of alternative process sequences considering the trade-off between schedule quality and the costs of process sequences. However, traditional techniques are not able to provide such flexibility for DM.

The above issues lead to flexible process sequencing problems (FPSs), which simultaneously considers alternative process plans with multiple objectives and the flexibility of process sequences. Over the past decade, a number of models have been developed to solve the process sequencing problems, but only few models [1, 7] have been reported to design the process sequencing problem considering the above issues. To date, solving the problem of flexible process sequencing with multiple objectives that are conflicting in nature is still a hard task.

2.2 Multi-objective Evolutionary Optimization

Assume all the objective functions F_m are to be minimized. Mathematically, multi-objective optimization problems (MOOPs) can be represented as the following vector mathematical programming problems:

$$Minimize \quad F(X) = \{F_1(X), F_2(X), ..., F_m(X)\}, \qquad (1)$$

where X denotes a solution and $F_m(X)$ is generally a nonlinear objective function. When the following inequalities hold between two solutions X_1 and X_2, X_2 is a *non-dominated solution* and is said to *dominate* $X_1 (X_2 \succ X_1)$:

$$\forall m : F_m(X_1) \geq F_m(X_2) \;\; and \;\; \exists n : F_n(X_1) > F_n(X_2). \quad (2)$$

When the following inequality hold between two solutions X_1 and X_2, X_2 is said to *weakly dominate* $X_1 (X_2 \succeq X_1)$:

$$\forall m : F_m(X_1) \geq F_m(X_2). \qquad (3)$$

A feasible solution X^* is said to be a *Pareto-optimal solution* if and only if there does not exist a feasible solution X where X dominates X^*. The corresponding vector of Pareto-optimal solutions is called *Pareto-optimal front*.

By making use of Pareto dominance relationship, multi-objective evolutionary algorithms (MOEAs) are capable of performing the fitness assignment of multiple objectives without using relative preferences of multiple objectives. Thus, all the objective functions can be optimized simultaneously. As a result, MOEA seems to be an alternative approach to solving production planning and inspection planning problems on the assumption that no prior domain knowledge is available.

3. PROBLEM STATEMENT

3.1 The FMS Environment

An FMS consists of a set of identical and/or complementary numerically controlled machines and tool systems. All components are connected through an AGV system. Figure 1 shows the layout of a simple FMS with several machines, AGVs and a tool system.

In order to design the production planning of FMSs, the environment within which the FMS under consideration operates can be described below.

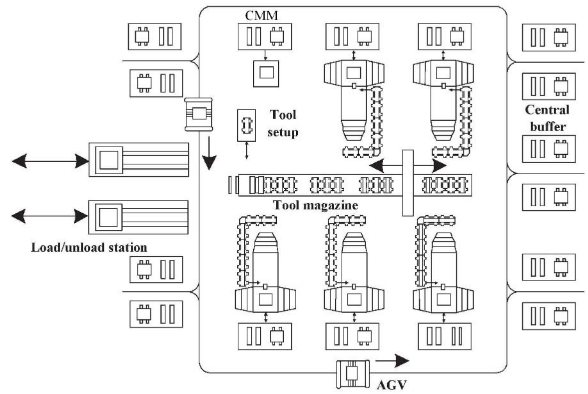

Figure 1: FMS with several machines, a coordinate measuring machine (CMM), AGVs and a central tool magazine.

- The term *machine* is to describe a machine cell. A machine cell consists of several identical devices/machines. The types and number of machines are known. There is a sufficient input/output buffer space at each machine.
- A *part* type requires a number of *operations*. A number of part types will be manufactured simultaneously in batches. Parts can choose one or more machines at each of their operation stages, and the transportation of the parts within different machines is handled by an AGV system.
- A machine can perform several types of operations, and an operation can be performed on alternative machines.
- A machine can only process an operation at one time. Operations to be performed in the machine are non-preemptive. Operation lot splitting is ignored in this paper.
- A *process sequence* is a series of machine indices corresponding to operations of all parts. Based on a process sequence, each operation is operated on its corresponding machine. An illustrative process sequence of 3 parts and 10 operations is presented in Figure 2, and the operations are operated on 3 different machines. An example of the series of machine indices to be optimized is Y=[1 1 1 3 1 2 2 2 3 3].
- Workload on each machine is contributed by those operations assigned to a machine.
- A load/unload (L/U) station serves as a distribution center for parts not yet processed and as a collection center for parts finished. All vehicles start from the L/U station initially and return to there after accomplishing all their assignments. There are sufficient input/output buffer spaces at the L/U station.
- The number of AGVs is given and the transportation time of AGVs are known. Some machines may not be linked.
- AGVs carry a limited number of products at a time. They move along predetermined paths, with the assumption of no delay because of congestion. Preemption of trips is not allowed.

- It is assumed that all the design, layout and set-up issues within FMS have already been resolved.
- Real-time issues, such as traffic control, congestion, machine failure or downtime, scraps, rework, and vehicle dispatches for battery changer are ignored here and left as issues to be considered during real-time control.

Part index	1	2	3
Operation index	1 2 3 4	1 2 3	1 2 3
Process Sequence (Machine index)	1 1 1 3	1 2 2	2 3 3

Figure 2: A process sequence of 3 parts and 10 operations, operated on 3 different machines. For example, the operation 4 of the part 1 is assigned to the machine 3.

3.2 Mathematical Formulation of FPSs

3.2.1 Notations

In order to formulate FPSs, the following notations are introduced:

- i : part index, $i = 1, 2, 3, ..., I$.
- j : operation index for part i, $j = 1, 2, 3, ..., J_i$.
- k, l : machine index $k, l = 1, 2, 3, ..., K$.
- Y : process sequence.
- pv_i : production volume (unit) for part i.
- pt_{ijk} : processing time per unit to perform operation j of part i using machine k.
- m_k : maximum workload of machine k.
- tw_k : workload in machine k, $tw_k = pt_{ijk} \times pv_i$.
- rtw_k : workload ratio in machine k, $rtw_k = \frac{tw_k}{m_k}$.
- ew : average workload of machines.
- $s_{ikl} : \begin{cases} 1, & \text{if part } i \text{ is to transfer from machine } k \text{ to } l; \\ 0, & \text{otherwise.} \end{cases}$
- $x_{ijk} : \begin{cases} 1, & \text{if machine } k \text{ is selected to perform} \\ & \text{operation } j \text{ of part } i; \\ 0, & \text{otherwise.} \end{cases}$
- abl : available capacity of AGV per trip, abl is set to 10 in this chapter.
- n_{ikl} : the number of trips between machine k and l for part i,
$$n_{ikl} = s_{ikl} \times \lceil \frac{pv_i}{abl} \rceil,$$
where the bracket represents a ceiling operation.
- tm_{kl} : transportation time from machine k to l. If machines k and l are not linked, it is set to be a negative value for constraint handling.
- t_{ikl} : total transportation time between machines k and l for part i,
$$t_{ikl} = n_{ikl} \times tm_{kl}.$$

3.2.2 Objectives

There are three objectives to be optimized in flexible process sequencing problems, described below.

1. Minimization of total flow time. This objective is to minimize the processing time and transportation time for producing the parts. The total machine processing time (e_1) is defined as Equation 4, the transportation time (e_2) is defined as Equation 5, and the total flow time (f_1) is defined as Equation 6. Transportation between unlinked machines are penalized in e_2.
$$e_1 = \sum_{i=1}^{I} \sum_{j=1}^{J_i} \sum_{k=1}^{K} pv_i \times pt_{ijk} \times x_{ijk}, \tag{4}$$
$$e_2 = \sum_{i=1}^{I} \sum_{j=1}^{J_i - 1} \sum_{k=1}^{K} \sum_{l=1}^{K} t_{ikl} \times x_{ijk} \times x_{i(j+1)l}, \tag{5}$$
$$f_1 = e_1 + e_2. \tag{6}$$
2. Minimization of machine workload unbalance. Balancing the machine workload can avoid creating bottleneck machines. The objective function (f_2) is defined as Equation 7.
$$f_2 = \sum_{k=1}^{K} (rtw_k - ew)^2. \tag{7}$$
3. Minimization of greatest machine workload. Pursuing this objective also implies attempting to minimize the total flow time. The objective function (f_3) is defined as Equation 8.
$$f_3 = max\{rtw_k\}. \tag{8}$$

3.2.3 Multi-objective Mathematical Model

The overall multi-objective mathematical model of FPSs can be formulated as follows. Given the production volume pv_i, the processing time pt_{ijk}, the maximum workload m_k, the available capacity of AGV per trip abl, the transportation time tm_{kl} and the tool costs c_{ijk}, find a series of machine indices, Y, for operations of all parts such that
$$minimize \quad f_1, f_2, f_3, \tag{9}$$
subject to
$$\sum_{k=1}^{K} x_{ijk} = 1, \quad \forall (i, j), \tag{10}$$
$$tm_{kl} \geq 0, \quad \forall (k, l), \tag{11}$$
$$rtw_k \leq 1, \quad \forall i. \tag{12}$$

The constraint, Equation 10, ensures that only one machine is selected for each operation of a part. Equation 11 ensures an AGV path exists between machines k and l. Equation 12 is to ensure the machine workload tw_k is smaller or equal to its maximum machine workload m_k.

If the total number of machines is x and the total number of operations is y, then the complexity of the investigated problem is $O(x^y)$.

4. MULTI-OBJECTIVE MEMETIC ALGORITHM WITH FITNESS INHERITANCE MEFI

4.1 Schemata-Guided Local Search Strategy

Based on schema theorem and the niche hypothesis [5], a schemata-guided local search strategy is proposed to be combined with MOGA for improving the convergence speed to the Pareto-front. Extended from the niche hypothesis, it is assumed that, given a MOOP with Q Pareto-optimal solutions, Q Pareto-optimal solutions can be regarded as Q niches of the MOOP. In the worst case, to ensure MOEAs is capable of searching Q Pareto-optimal solutions, it is assumed that the population were divided into Q species (subpopulations). Thus, each species is expect to optimize its own niche (Pareto-optimal solution), as shown in Figure 3. Therefore, the optimal schemata of a species is its Pareto-optimal solution.

Let the schema of species be H_q, where the fixed positions are the maximum common string of all individuals in its species and the others are "don't care"(*). Since species are in the same population, a schemata of a species may be disrupted by schemata of the other species due to genetic operators. The disruption between species can be further classified into the following two types:

1. **Species disrupt noise**: The fixed schemata of H_{origin} are altered to "don't care" schemata by the corresponding positions of the schemata H_{other}. Thus, a species requires more time for fixing it's "don't care" schemata.

2. **Species hitchhiking noise**: The "don't care" schemata of H_{origin} are altered to fixed schema by the corresponding positions of the schema H_{other}. If the altered schemata are located in the similarity regions of their optimal schemata, the change is good for the schemata H_{origin}. On the contrary, the change is bad for the schemata H_{origin}.

Based on the foregoing inference, it is desired that a species should keep its good schemata (building blocks) while making good efforts to alter its "don't care" schemata to its ideal optimal schemata. As results, a schemata-guided local search strategy is proposed based on this guideline. Information of fixed and "don't care" schemata in species are utilized to guide local search. However, the key question of this local search strategy is that how do we classify population to different species when true Pareto-optimal solutions of MOOPs are unknown. To deal with this question, it is assumed that the best individuals in each objective functions are the *pioneers* of each species. These pioneers will be used to classify all individuals in population to different species.

Given a maximum local search times $MaxLS$ and a temporary elite set E', the procedure of the used schemata-guided local search strategy is written as follows:

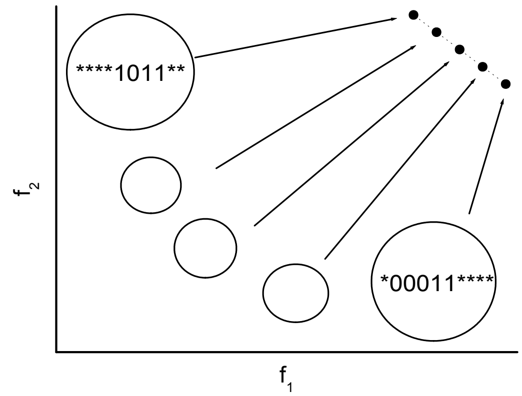

Figure 3: The population were divided into several species, and each species optimizes its own niche (Pareto-optimal solution).

Step 1: (Identification) Identify the best individuals $B_q, q = 1, 2, ..., Q$, in each objective from the current population. For FPSs, Q=3.

Step 2: (Classification) Classify the current population into Q species by the best solutions in each objective.

Step 3: (Schemata computation) For each species, compute its schemata H_q. Both fixed and "don't care" schemata are identified.

Step 4: (Parameter setting) Let $q = 1, counter = 0$.

Step 5: (Perturbation) Perturb B_q into a new solution B'_q. According to H_q, apply the mutation operator only on "don't care" locations of B_q with a mutation probability p_m.

Step 6: (Evaluation) Evaluate the objective functions of B'_q. Let $counter = counter + 1$.

Step 7: (Comparison) There is 3 cases in comparisons of B_q and B'_q. **Case 1**: If B_q dominates B'_q and $counter < MaxLS$, go to Step 5. **Case 2**: If B_q is dominated by B'_q, replace B_q by B'_q. **Case 3**: If B_q and B'_q doesn't dominated each other. Stored B'_q in a temporary elite set E'.

Step 8: (Termination test) Let $q = q + 1$ and counter=0, if q>Q, stop the local search strategy. Otherwise, go to Step 5.

4.2 Fitness Inheritance

An efficiency enhancement techniques called fitness inheritance [2] is used for speedup of MEFI. During the evolution of EAs, the fitness of some proportion of individuals in the subsequent population is inherited. This proportion is called the inheritance proportion, p_i.

Mathematically, for a multi-objective problem with z objective, the used fitness inheritance is defined as

$$f_z = \frac{w_1 f_{z,p1} + w_2 f_{z,p2}}{w_1 + w_2}, \tag{13}$$

where f_z is the fitness value in objective z, w_1, w_2 are the weights for the two parents p_1, p_2, and $f_(z, p_1)$, $f_(z, p_2)$ is the fitness values of p_1, p_2 in objective z, respectively. In this paper, w_1 and w_2 are set to 1.

According the literature of fitness inheritance, the population size of FIEA should be bigger than the population size used for MOGA, as shown in the following equation:

$$N_{pop,FIEA} = \frac{N_{pop,MOGA}}{1 - p_i^3} \tag{14}$$

4.3 MEFI for solving FPSs

4.3.1 Representation and Operators

A series of machine indices Y for operations of all parts is directly encoded as a integer chromosome. The range of each gene of Y is [1, K]. Each gene of Y stands for a machine index.

The selection operator of MEFI uses a binary tournament selection which works as follows. Choose two individuals randomly from the population and copy the better individual into the intermediate population. The one-point crossover is used in MEFI. A simple mutation operator is used to alter genes. For each gene, randomly generate a real value from the range [0, 1] with the probability p_m.

MEFI uses a generalized Pareto-based scale-independent fitness function GPSIFF [6] by the following function:

$$F(X) = p - q + c, \tag{15}$$

where p is the number of individuals which can be dominated by the individual X, and q is the number of individuals which can dominate the individual X in the objective space. c is the number of all participant individuals.

Based on the proposed chromosome representation, Equation 10 is always satisfied. If Equation 11 is violated, the transportation time between machines k and l, tm_{kl}, is set to be a large value, 10^7. In this way, f_2 will be penalized. For each machine k, if Equation 12 is not satisfied, one is added to r_{twk}, as follows:

$$r_{twk} = \begin{cases} \frac{tw_k}{m_k}, & \text{if } tw_k \leq m_k; \\ \frac{tw_k}{m_k} + 1, & \text{otherwise.} \end{cases} \tag{16}$$

4.4 Procedure of MEFI

Since it has been recognized that the incorporation of elitism may be useful in maintaining diversity and improving the performance of multi-objective EAs [3], MEFI selects a number of elitists from an elite set E in the selection step. The elite set E with capacity E_{max} maintains the best non-dominated solutions generated so far. In addition, an external set $\overline{E}$ with no capacity is used to store all the non-dominated solutions ever generated so far. The procedure of MEFI is written as follows:

Step 1: (Initialization) Randomly generate an initial population of N_{pop} individuals and create two empty elite sets E, $\overline{E}$ and an empty temporary elite set E'.

Step 2: (Evaluation) For each individual Y in the population, excluding the inherited individuals, compute the value of objective functions $f_1(Y)$, $f_2(Y)$, and $f_3(Y)$.

Step 3: (Fitness assignment) Assign each individual a fitness value by using GPSIFF.

Table 1: The parameter settings of MEFI and MOGA.

Parameters	MEFI	MOGA
N_{pop}	115	100
E_{max}	115	100
p_s	0.25	0.25
p_i	0.5	N/A
p_c	0.6	0.6
p_m	0.05	0.05
$MaxLS$	3	N/A

Step 4: (Local search) Apply the proposed schemata-guided local search strategy. Non-dominated solutions obtained by the local search strategy will be stored in temporary elite set E'.

Step 5: (Update elite sets) Add the non-dominated individuals in both the population and E' to E, and empty E'. Considering all individuals in E, remove the dominated ones in E. Add E to $\overline{E}$, remove the dominated ones in $\overline{E}$. If the number of non-dominated individuals in E is larger than E_{max}, randomly discard excess individuals.

Step 6: (Selection) Select $N_{pop} - N_{ps}$ individuals from the population using the binary tournament selection and randomly select N_{ps} individuals from E to form a new population, where $N_{ps} = N_{pop} \times p_s$ and p_s is a selection proportion. If N_{ps} is greater than the number N_E of individuals in E, let $N_{ps} = N_E$.

Step 7: (Recombination) Perform the one-point crossover operation with a recombination probability p_c.

Step 8: (Fitness inheritance) Perform fitness inheritance on the selected $N_{pop} \times p_i$ individuals. The inherited objective values are calculated according to Equation 13.

Step 9: (Mutation) Apply the mutation operator to each gene in the individuals with a mutation probability p_m.

Step 10: (Termination test) If a stopping condition is satisfied, stop the algorithm and output $\overline{E}$. Otherwise, go to Step 2.

5. RESULTS AND DISCUSSION

Six benchmark problems: *m3o10*, *m4o20*, *m5o100*, *m5o200*, *m10o100* and *m10o200*, where *mxoy* stands for the x machine and y operation problem. A MOGA, MEFI without the local search strategy and fitness inheritance, is implemented to solve FPSs as the baseline performance. The parameter settings of MEFI and MOGA are given in Table 1. Thirty independent runs with the same number of function evaluations $100xy$ were performed per test problems.

The coverage metric $C(A, B)$ of two solution sets A and B [8] used to compare the performance of two corresponding algorithms considering the six objectives:

$$C(A, B) = \frac{|\{a \in A, b \in B, a \succeq b\}|}{|B|}, \tag{17}$$

Fig. 4 depicts the coverage metrics of $C(MEFI, MOGA)$ and $C(MOGA, MEFI)$ from 30 runs. In solving the small problem *m3o10*, Fig. 4 shows that the performance of MEFI

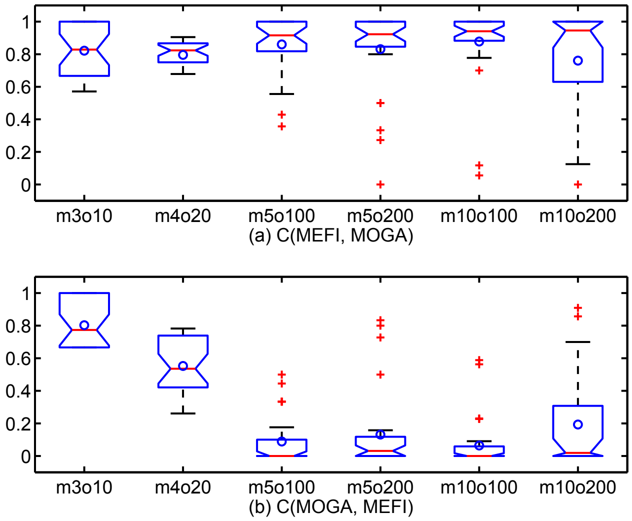

Figure 4: Box plots based on the cover metric. (a) C(MEFI, MOGA), (b) C(MOGA, MEFI).

and MOGA are almost the same. For another small problem *m4o20*, the non-dominated solutions obtained by MEFI dominates 80% of the solutions obtained by MOGA in average, while the non-dominated solutions obtained by MOGA only dominates 60% of the non-dominated solutions obtained by MEFI in average. As the complexity of problems increases, Fig. 4 shows that 80%-90% of the non-dominated solutions obtained by MOGA are weakly dominated by the non-dominated solutions obtained by MEFI in solving the problems *m4o20*, *m5o100*, *m5o200*, *m10o100* and *m10o200*. On the contrast, the non-dominated solutions of MOGA dominate nearly 3-10% of the non-dominated solutions obtained by MEFI. Fig. 5 shows the non-dominated solutions obtained by thirty runs of MEFI and MOGA in solving the *m10o200* problem. The results indicate that MEFI can converge to better solutions more quickly than MOGA. It reveals that the proposed schemata-guided local search strategy and fitness inheritance plays an important role in obtaining good solutions and accelerating the convergence speed.

6. CONCLUSION

In this paper, a novel approach to solve flexible process sequencing problems using an multi-objective memetic algorithm MEFI is proposed. A schemata-guided local search strategy and fitness inheritance are integrated in the proposed algorithm for enhancing the performance. Experimental results demonstrated that the quality of non-dominated solutions obtained by MEFI is better than that of MOGA in terms of convergence speed and accuracy using the same number of function evaluations. While prior domain knowledge for the decomposition of problems or relative preferences of multiple objectives are not available, the proposed approach is an expedient method to solve flexible process sequencing problems. Moreover, the proposed approach can obtain a set of non-dominated solutions for decision makers in a single run. Decision makers can easily distinguish between the costs of different process sequences and choose more than one satisfactory process sequences at a time.

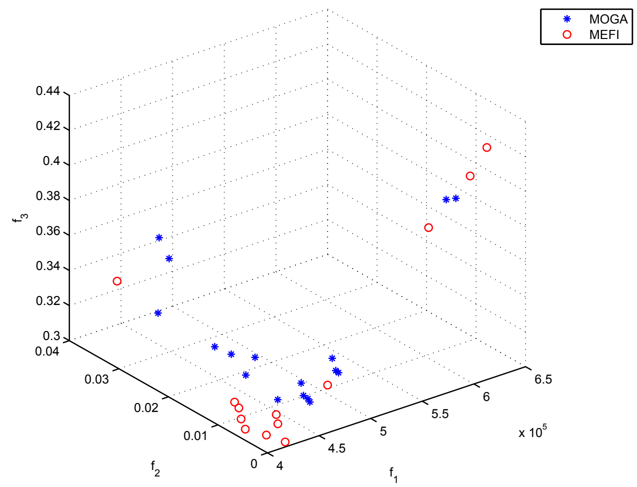

Figure 5: The non-dominated solutions obtained by MEFI and MOGA in solving the *m10o200* problem, merged from 30 runs.

7. ACKNOWLEDGMENTS

This work was supported by the National Science Council of Taiwan, R.O.C. under Contract NSC-96-2221-E-216-037-MY2 and NSC-095-SAF-I-564-616-TMS, and Chung-Hua University under Contract CHU-96-2221-E-216-037-MY2.

8. REFERENCES

[1] P. Brandimarte. Exploiting process plan flexibility in production scheduling: A multi-objective approach. *European Journal of Operational Research*, (114):59–71, 1999.

[2] J.-H. Chen, D. E. Goldberg, S.-Y. Ho, and K. Sastry. Fitness inheritance in multi-objective optimization. In *GECCO '02: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 319–326, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[3] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley-Interscience series in systems and optimization. John Wiley & Sons, 2001.

[4] M. Gen and R. Cheng. *Genetic algorithms and engineering design*. John Wiley, New York, 1997. 1944- Mitsuo Gen, Runwei Cheng. ill. ; 24 cm.

[5] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Pub. Co., 1989.

[6] S.-Y. Ho, L.-S. Shu, and J.-H. Chen. Intelligent evolutionary algorithms for large parameter optimization problems. *IEEE Transaction on Evolutionary Computation*, 8(6):522–541, Dec. 2004.

[7] C. Moon, Y.-Z. Li, and M. Gen. Evolutionary algorithm for flexible process sequencing with multiple objectives. In *Proceeding of IEEE International Conference on Computational Intelligence*, pages 27–32, 1998.

[8] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strengthen Pareto approach. *IEEE Transaction on Evolutionary Computation*, 4(3):257–271, 1999.

Memetic Approach for Multi-objective Flexible Process Sequencing Problems

Jian-Hung Chen
Department of Computer Science and Information Engineering
Chung-Hua University, Hsin-Chu, Taiwan 300
E-mail: jh.chen@ieee.org

Abstract—*This paper describes a novel multi-objective memetic algorithm for solving multi-objective flexible process sequencing problems in flexible manufacturing systems (FMSs). FMS can be described as an integrated manufacturing system consisting of machines, computers, robots, tools, and automated guided vehicles (AGVs). While FMSs give great advantages through the flexibility, FMSs usually pose complex problems on process sequencing of operations among multiple parts. Considering the machining time of operations and machine workload load balancing, the problem is formulated as multi-objective flexible process sequencing problems (FPSs). An efficient multi-objective memetic algorithm with fitness inheritance mechanism is proposed to solve FPSs. The experimental results demonstrate that our approach can efficiently solve FPSs and fitness inheritance can speed up the convergence speed of the proposed algorithm in solving FPSs.*

Keywords—process planning, flexible manufacturing systems, multi-objective optimization, memetic algorithms, fitness inheritance

1 Introduction

Computer-aided process planning (CAPP) is an automated system for preparation of a plan that specifies machines, machine conditions, operations, operation sequence, and tools required to production these components [1]. CAPP techniques are being developed in an attempt to overcome some of the problems occurring in manual process planning, such as long turn around times, inconsistent routing or tooling, non-uniqueness in cost and labor requirements and scarcity of skilled process planners. During the past two decades, a number of CAPP systems have been developed for the automated planning and increased efficiency of process planning function. Traditionally, the process sequencing has been solved by either the experience of process planners or a fixed and static process plan consisting of an ordered sequence of operations [2]. However, the traditional mythologies are not suitable in real flexible environment, because the techniques have a few constraints in order to cope with dynamic situations of the flexible environment [3]. Moreover, as the number of operations increase, it poses more difficulties for decision makers to plan a cost-effective process sequences for manufacturing.

In this paper, a memetic algorithm using fitness inheritance (MAFI) is proposed to solve multi-objective flexible process sequencing problems (FPSs) having three objectives: minimizing total machining time, maximum machine workload and machine workload unbalance. The fundamental difference of the proposed approach from the traditional approaches is that the problem decomposition and relative preferences are not necessary. In addition, the proposed approach can obtain a set of non-dominated solutions for decision makers in a single run. Decision makers can easily distinguish between the costs of different process sequences and choose more than one satisfactory process sequences at a time. Six benchmark problems with different complexities are used to evaluate the performance of the proposed approach. A multi-objective genetic algorithm (MOGA) without local search and fitness inheritance is used for performance comparisons. It is shown empirically that MAFI outperforms MOGA in terms of the solution quality.

This paper is organized as follows: Section 2 presents the background of process sequencing problems, multi-objective optimization problems and evolutionary algorithms. Section 3 introduces the setup of flexible manufacturing system and the mathematical formulation of FPSs. Section 4 presents the multi-objective memetic algorithm for solving FPSs. Section 5 presents the experimental analysis of the proposed algorithm, and Section 6 summarizes our conclusions.

2 Background

2.1 Process Sequencing Problems

Flexible process sequencing problems are well known among the combinatorial optimization problems. Previous research focused on two important key issues of process sequencing problems, described as follows. The first key issue is the objective functions of process sequencing. Several approaches are proposed for process sequencing with various objectives. For examples, Kusiak and Finke [2] have developed a model for selecting a set of process plans with the objective of minimizing the makespan. Bhaskaran [4] provided a model for minimizing the total machine time and the total number of processing steps. Zhang and Huang [5] presented a fuzzy-based model for the selection of a set of process plans considering the imprecise information of shop floor. Furthermore, various heuristic approaches [6] have been proposed for minimizing the makespan.

Another key issue that arises recently is the alternative process sequences. In the view of real time scheduling, alternative process sequences provide additional capability for the decision maker (DM) to cope with unpredictable events such as machine failures or rush orders. From the view of off-line scheduling, alternative process sequences may be used to improve the schedule quality by reducing the load on bottleneck machines [4]. Generally speaking, finding a set of optimal alternative process sequences economically plays an important role in solving the process sequencing problems. However, it is easier to obtain the alternative process sequences with single objective than that with multiple objectives. It is because, simultaneous optimization of several incommensurable and conflicting objectives in nature is much more complex and difficult. On the other hand, flexible process sequencing with multiple objectives makes more practical applications in the design phase of industrial manufacturing. As a result, it is essential but also a challenge for DM to prepare a set of alternative process sequences considering the trade-off between schedule quality and the costs of process sequences.

The above issues lead to flexible process sequencing problems (FPSs), which simultaneously considers alternative process plans with multiple objectives and the flexibility of process sequences. Over the past decade, a number of models have been developed to solve the process sequencing problems, but only few models [3], [4] have been reported to design the process sequencing problem considering the above issues. To date, solving the problem of flexible process sequencing with multiple objectives that are conflicting in nature is still a hard task.

2.2 Multi-objective Evolutionary Optimization

Assume all the objective functions F_m are to be minimized. Mathematically, multi-objective optimization problems (MOOPs) can be represented as the following vector mathematical programming problems:

$$Minimize \quad F(X) = \{F_1(X), F_2(X), ..., F_m(X)\}, \quad (1)$$

where X denotes a solution and $F_m(X)$ is generally a nonlinear objective function. When the following inequalities hold between two solutions X_1 and X_2, X_2 is a *non-dominated solution* and is said to *dominate* $X_1 (X_2 \succ X_1)$:

$$\forall m : F_m(X_1) \geq F_m(X_2) \quad and \quad \exists n : F_n(X_1) > F_n(X_2). \quad (2)$$

When the following inequality hold between two solutions X_1 and X_2, X_2 is said to *weakly dominate* $X_1 (X_2 \succeq X_1)$:

$$\forall m : F_m(X_1) \geq F_m(X_2). \quad (3)$$

A feasible solution X^* is said to be a *Pareto-optimal solution* if and only if there does not exist a feasible solution X where X dominates X^*. The corresponding vector of Pareto-optimal solutions is called *Pareto-optimal front*.

In the past few years, multi-objective evolutionary algorithms (MOEAs) have been recognized to be well-suited for solving MOOPs because their abilities to exploit and explore multiple solutions in parallel and to find a widespread set of non-dominated solutions in a single run [7]. By making use of Pareto dominance relationship, MOEAs are capable of performing the fitness assignment of multiple objectives without using relative preferences of multiple objectives. Thus, all the objective functions can be optimized simultaneously. One of the recent growing areas in evolutionary algorithms (EAs) research is memetic agorithms (MAs). MAs are population-based meta-heuristic search methods inspired by Darwinian principles of natural evolution and Dawkins notion of a meme defined as a unit of cultural evolution that is capable of local refinements [8]. From an optimization point of view, MAs are hybrid EAs that combine global and local search by using an EA to perform exploration while the local search method performs exploitation. Combining global and local search is known as an efficient strategy in many successful optimization approaches [9], [10].

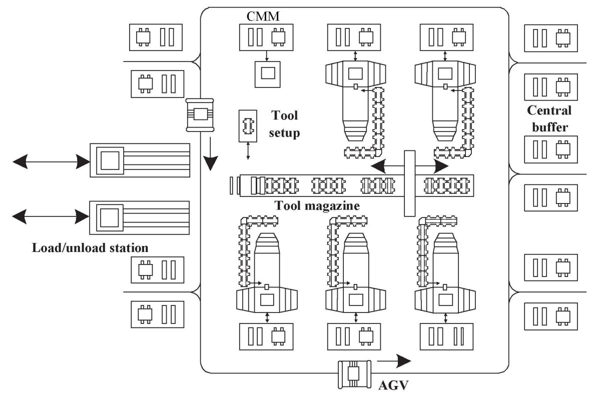

Fig. 1. FMS with several machines, a coordinate measuring machine (CMM), AGVs and a central tool magazine.

3 Problem Statement

The aim of flexible process sequencing is to develop a cost-effective and operative process sequences for the assignments of operation to machines over planning phases. With the assignments of operations to machines, three optimization objectives: minimizing total machining time, machine workload unbalance, and greatest machine workload are considered in this paper.

3.1 The FMS Environment

An FMS consists of a set of identical and/or complementary numerically controlled machines and tool systems. All components are connected through an AGV system. Figure 1 shows the layout of a simple FMS with several machines, AGVs and a tool system.

In order to design the production planning of FMSs, the environment within which the FMS under consideration operates can be described below.

Part index	1	2	3
Operation index	1 2 3 4	1 2 3	1 2 3
Process Sequence (Machine index)	1 1 1 3	1 2 2	2 3 3

Fig. 2. A process sequence of 3 parts and 10 operations, operated on 3 different machines. For example, the operation 4 of the part 1 is assigned to the machine 3.

- The term *machine* is to describe a machine cell. A machine cell consists of several identical devices/machines. The types and number of machines are known. There is a sufficient input/output buffer space at each machine.
- A *part* type requires a number of *operations*. A number of part types will be manufactured simultaneously in batches. Parts can choose one or more machines at each of their operation stages, and the transportation of the parts within different machines is handled by an AGV system.
- A machine can perform several types of operations, and an operation can be performed on alternative machines.
- A machine can only process an operation at one time. Operations to be performed in the machine are non-preemptive. Operation lot splitting is ignored in this paper.
- A *process sequence* is a series of machine indices corresponding to operations of all parts. Based on a process sequence, each operation is operated on its corresponding machine. An illustrative process sequence of 3 parts and 10 operations is presented in Figure 2, and the operations are operated on 3 different machines. An example of the series of machine indices to be optimized is Y=[1 1 1 3 1 2 2 2 3 3].
- Workload on each machine is contributed by those operations assigned to a machine.
- A load/unload (L/U) station serves as a distribution center for parts not yet processed and as a collection center for parts finished. All vehicles start from the L/U station initially and return to there after accomplishing all their assignments. There are sufficient input/output buffer spaces at the L/U station.
- The number of AGVs is given and the transportation time of AGVs are known. Some machines may not be linked.
- AGVs carry a limited number of products at a time. They move along predetermined paths, with the assumption of no delay because of congestion. Preemption of trips is not allowed.
- It is assumed that all the design, layout and set-up issues within FMS have already been resolved.
- Real-time issues, such as traffic control, congestion, machine failure or downtime, scraps, rework, and vehicle dispatches for battery changer are ignored here and left as issues to be considered during real-time control.

3.2 Mathematical Formulation of FPSs

3.2.1 Notations: In order to formulate FPSs, the following notations are introduced:

- i : part index, $i = 1, 2, 3, ..., I$.
- j : operation index for part i, $j = 1, 2, 3, ..., J_i$.
- k, l : machine index $k, l = 1, 2, 3, ..., K$.
- Y : process sequence.
- pv_i : production volume (unit) for part i.
- pt_{ijk} : processing time per unit to perform operation j of part i using machine k.
- m_k : maximum workload of machine k.
- tw_k : workload in machine k, $tw_k = pt_{ijk} \times pv_i$.
- rtw_k : workload ratio in machine k, $rtw_k = \frac{tw_k}{m_k}$.
- ew : average workload of machines.
- s_{ikl} : $\begin{cases} 1, & \text{if part } i \text{ is to transfer from machine } k \text{ to } l; \\ 0, & \text{otherwise.} \end{cases}$
- x_{ijk} : $\begin{cases} 1, & \text{if machine } k \text{ is selected to perform} \\ & \text{operation } j \text{ of part } i; \\ 0, & \text{otherwise.} \end{cases}$
- abl : available capacity of AGV per trip, abl is set to 10 in this chapter.
- n_{ikl} : the number of trips between machine k and l for part i,
$$n_{ikl} = s_{ikl} \times \lceil \frac{pv_i}{abl} \rceil,$$
where the bracket represents a ceiling operation.
- tm_{kl} : transportation time from machine k to l. If machines k and l are not linked, it is set to be a negative value for constraint handling.
- t_{ikl} : total transportation time between machines k and l for part i,
$$t_{ikl} = n_{ikl} \times tm_{kl}.$$

3.2.2 Objectives: There are three objectives to be optimized in flexible process sequencing problems, described below.

1) Minimization of total flow time. This objective is to minimize the processing time and transportation time for producing the parts. The total machine processing time (e_1) is defined as Equation 4, the transportation time (e_2) is defined as Equation 5, and the total flow time (f_1) is defined as Equation 6. Transportation between unlinked machines are penalized in e_2.

$$e_1 = \sum_{i=1}^{I} \sum_{j=1}^{J_i} \sum_{k=1}^{K} pv_i \times pt_{ijk} \times x_{ijk}, \tag{4}$$

$$e_2 = \sum_{i=1}^{I} \sum_{j=1}^{J_i - 1} \sum_{k=1}^{K} \sum_{l=1}^{K} t_{ikl} \times x_{ijk} \times x_{i(j+1)l}, \tag{5}$$

$$f_1 = e_1 + e_2. \tag{6}$$

2) Minimization of machine workload unbalance. Balancing the machine workload can avoid creating bottleneck

machines. The objective function (f_2) is defined as Equation 7.

$$f_2 = \sum_{k=1}^{K} (rtw_k - ew)^2. \tag{7}$$

3) Minimization of greatest machine workload. Pursuing this objective also implies attempting to minimize the total flow time. The objective function (f_3) is defined as Equation 8.

$$f_3 = max\{rtw_k\}. \tag{8}$$

3.2.3 Multi-objective Mathematical Model: The overall multi-objective mathematical model of FPSs can be formulated as follows. Given the production volume pv_i, the processing time pt_{ijk}, the maximum workload m_k, the available capacity of AGV per trip abl, the transportation time tm_{kl} and the tool costs c_{ijk}, find a series of machine indices, Y, for operations of all parts such that

$$minimize \quad f_1, f_2, f_3, \tag{9}$$

subject to

$$\sum_{k=1}^{K} x_{ijk} = 1, \quad \forall (i,j), \tag{10}$$

$$tm_{kl} \geq 0, \quad \forall (k,l), \tag{11}$$

$$rtw_k \leq 1, \quad \forall i. \tag{12}$$

The constraint, Equation 10, ensures that only one machine is selected for each operation of a part. Equation 11 ensures an AGV path exists between machines k and l. Equation 12 is to ensure the machine workload tw_k is smaller or equal to its maximum machine workload m_k.

If the total number of machines is x and the total number of operations is y, then the complexity of the investigated problem is $O(x^y)$.

4 Multi-objective Memetic Algorithm with Fitness Inheritance MAFI

The proposed MAFI differs from MOGA in the local search strategy and fitness inheritance. The used schemata-guided local search strategy is presented in Section 4.1. Fitness inheritance is summarized in Section 4.2. MAFI for solving FPSs is presented in Section 4.3, including the representation of chromosomes, genetic operators, constraint handling, and the procedure of MAFI.

4.1 Schemata-Guided Local Search Strategy

Based on schema theorem and the niche hypothesis [11], a schemata-guided local search strategy is proposed to be combined with MOGA for improving the convergence speed to the Pareto-front. Extended from the niche hypothesis, it is assumed that, given a MOOP with Q Pareto-optimal solutions, Q Pareto-optimal solutions can be regarded as Q niches of the MOOP. In the worst case, to ensure MOEAs is capable

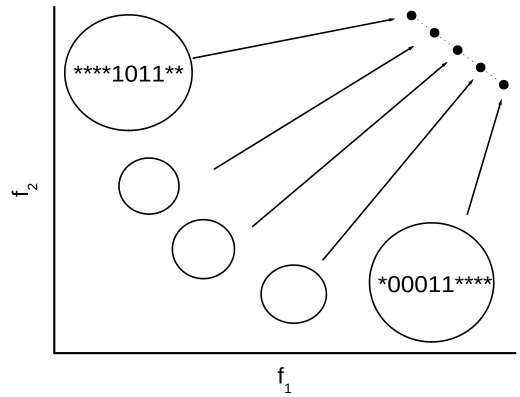

Fig. 3. The population were divided into several species, and each species optimizes its own niche (Pareto-optimal solution).

of searching Q Pareto-optimal solutions, it is assumed that the population were divided into Q species (sub-populations). Thus, each species is expect to optimize its own niche (Pareto-optimal solution), as shown in Figure 3. Therefore, the optimal schemata of a species is its Pareto-optimal solution.

Let the schema of species be H_q, where the fixed positions are the maximum common string of all individuals in its species and the others are "don't care"(*). Since species are in the same population, a schemata of a species may be disrupted by schemata of the other species due to genetic operators. The disruption between species can be further classified into the following two types:

1) **Species disrupt noise**: The fixed schemata of H_{origin} are altered to "don't care" schemata by the corresponding positions of the schemata H_{other}. Thus, a species requires more time for fixing it's "don't care" schemata.
2) **Species hitchhiking noise**: The "don't care" schemata of H_{origin} are altered to fixed schema by the corresponding positions of the schema H_{other}. If the altered schemata are located in the similarity regions of their optimal schemata, the change is good for the schemata H_{origin}. On the contrary, the change is bad for the schemata H_{origin}.

Based on the foregoing inference, it is desired that a species should keep its good schemata (building blocks) while making good efforts to alter its "don't care" schemata to its ideal optimal schemata. As results, a schemata-guided local search strategy is proposed based on this guideline. Information of fixed and "don't care" schemata in species are utilized to guide local search. However, the key question of this local search strategy is that how do we classify population to different species when true Pareto-optimal solutions of MOOPs are unknown. To deal with this question, it is assumed that the best individuals in each objective functions are the *pioneers* of each species. These pioneers will be used to classify all individuals in population to different species.

Given a maximum local search times $MaxLS$ and a temporary elite set E', the procedure of the used schemata-guided local search strategy is written as follows:

Step 1 : (Identification) Identify the best individuals $B_q, q = 1, 2, ..., Q$, in each objective from the current population. For FPSs, Q=3.

Step 2 : (Classification) Classify the current population into Q species by the best solutions in each objective.

Step 3 : (Schemata computation) For each species, compute its schemata H_q. Both fixed and "don't care" schemata are identified.

Step 4 : (Parameter setting) Let $q = 1, counter = 0$.

Step 5 : (Perturbation) Perturb B_q into a new solution B'_q. According to H_q, apply the mutation operator only on "don't care" locations of B_q with a mutation probability p_m.

Step 6 : (Evaluation) Evaluate the objective functions of B'_q. Let $counter = counter + 1$.

Step 7 : (Comparison) There is 3 cases in comparisons of B_q and B'_q. **Case 1**: If B_q dominates B'_q and $counter < MaxLS$, go to Step 5. **Case 2**: If B_q is dominated by B'_q, replace B_q by B'_q. **Case 3**: If B_q and B'_q doesn't dominated each other. Stored B'_q in a temporary elite set E'.

Step 8 : (Termination test) Let $q = q + 1$ and counter=0, if q¿Q, stop the local search strategy. Otherwise, go to Step 5.

4.2 Fitness Inheritance

An efficiency enhancement techniques called fitness inheritance [12] is used for speedup of MAFI. During the evolution of EAs, the fitness of some proportion of individuals in the subsequent population is inherited. This proportion is called the inheritance proportion, p_i.

Mathematically, for a multi-objective problem with z objective, the used fitness inheritance is defined as

$$f_z = \frac{w_1 f_{z,p1} + w_2 f_{z,p2}}{w_1 + w_2}, \tag{13}$$

where f_z is the fitness value in objective z, w_1, w_2 are the weights for the two parents p_1, p_2, and $f_(z, p_1)$, $f_(z, p_2)$ is the fitness values of p_1, p_2 in objective z, respectively. In this paper, w_1 and w_2 are set to 1.

According the literature of fitness inheritance, the population size of FIEA should be bigger than the population size used for MOGA, as shown in the following equation:

$$N_{pop,FIEA} = \frac{N_{pop,MOGA}}{1 - p_i^3} \tag{14}$$

4.3 MAFI for solving FPSs

A series of machine indices Y for operations of all parts is directly encoded as a integer chromosome. The range of each gene of Y is [1, K]. Each gene of Y stands for a machine index.

The selection operator of MAFI uses a binary tournament selection which works as follows. Choose two individuals randomly from the population and copy the better individual into the intermediate population. Crossover is a recombination process in which genes from two selected parents are recombined to generate offspring chromosomes. The one-point crossover is used in MAFI. A simple mutation operator is used to alter genes. For each gene, randomly generate a real value from the range [0, 1]. If the value is smaller than the mutation probability p_m, replace its index with a randomly generated integer among its possible values.

MAFI uses a generalized Pareto-based scale-independent fitness function GPSIFF [13] by the following function:

$$F(X) = p - q + c, \tag{15}$$

where p is the number of individuals which can be dominated by the individual X, and q is the number of individuals which can dominate the individual X in the objective space. c is the number of all participant individuals.

Based on the proposed chromosome representation, Equation 10 is always satisfied. If Equation 11 is violated, the transportation time between machines k and l, tm_{kl}, is set to be a large value, 10^7. In this way, f_2 will be penalized. For each machine k, if Equation 12 is not satisfied, one is added to r_{twk}, as follows:

$$r_{twk} = \begin{cases} \frac{tw_k}{m_k}, & \text{if } tw_k \le m_k; \\ \frac{tw_k}{m_k} + 1, & \text{otherwise.} \end{cases} \tag{16}$$

4.4 Procedure of MAFI

Since it has been recognized that the incorporation of elitism may be useful in maintaining diversity and improving the performance of multi-objective EAs [7], MAFI selects a number of elitists from an elite set E in the selection step. The elite set E with capacity E_{max} maintains the best non-dominated solutions generated so far. In addition, an external set $\overline{E}$ with no capacity is used to store all the non-dominated solutions ever generated so far. The procedure of MAFI is written as follows:

Step 1 : (Initialization) Randomly generate an initial population of N_{pop} individuals and create two empty elite sets E, $\overline{E}$ and an empty temporary elite set E'.

Step 2 : (Evaluation) For each individual Y in the population, excluding the inherited individuals, compute the value of objective functions $f_1(Y)$, $f_2(Y)$, and $f_3(Y)$.

Step 3 : (Fitness assignment) Assign each individual a fitness value by using GPSIFF.

Step 4 : (Local search) Apply the proposed schemata-guided local search strategy. Non-dominated solutions obtained by the local search strategy will be stored in temporary elite set E'.

Step 5 : (Update elite sets) Add the non-dominated individuals in both the population and E' to E, and empty E'. Considering all individuals in E, remove the dominated ones in E. Add E to $\overline{E}$, remove the dominated ones in $\overline{E}$. If the number of non-dominated individuals in E is larger than E_{max}, randomly discard excess individuals.

Step 6 : (Selection) Select $N_{pop} - N_{ps}$ individuals from the population using the binary tournament selection and randomly select N_{ps} individuals from E to form a new population, where $N_{ps} = N_{pop} \times p_s$ and p_s is a selection proportion. If N_{ps} is greater than the number N_E of individuals in E, let $N_{ps} = N_E$.

Step 7 : (Recombination) Perform the one-point crossover operation with a recombination probability p_c.

Step 8 : (Fitness inheritance) Perform fitness inheritance on the selected $N_{pop} \times p_i$ individuals. The inherited objective values are calculated according to Equation 13.

Step 9 : (Mutation) Apply the mutation operator to each gene in the individuals with a mutation probability p_m.

Step 10 :(Termination test) If a stopping condition is satisfied, stop the algorithm and output $\overline{E}$. Otherwise, go to Step 2.

5 Results and discussion

Considering the real manufacturing environment, we derived the AGV transportation time matrix and six benchmark problems: *m3o10*, *m4o20*, *m5o100*, *m5o200*, *m10o100* and *m10o200*, where *mxoy* stands for the x machine and y operation problem. In order to further investigate the performance of MAFI, a MOGA (MAFI without the local search strategy and fitness inheritance) is also implemented to solve FPSs. The solutions obtained by MOGA are used as the baseline performance for comparisons. The parameter settings of MAFI and MOGA are given in Table I. All the parameters of MAFI and MOGA in each experiment are the same. Thirty independent runs were performed per test problems, compared with the same number of function evaluations $100xy$.

The coverage metric $C(A, B)$ of two solution sets A and B [14] used to compare the performance of two corresponding algorithms considering the six objectives:

$$C(A, B) = \frac{|\{a \in A, b \in B, a \succeq b\}|}{|B|}, \quad (17)$$

where $\succeq$ stands for weakly dominate in Pareto dominance relationship. The value $C(A, B) = 1$ means that all individuals in B are weakly dominated by A. On the contrary, $C(A, B) = 0$ denotes that none of individuals in B is weakly dominated by A. Because the C measure considers the weakly dominance relationship between two sets A and B, $C(A, B)$ is not necessarily equal to $1 - C(B, A)$. The comparison results of two solution sets using the coverage metric are depicted using box plots. A box plot provides an excellent visual result of a distribution. The box stretches from the lower hinge (defined as the 25th percentile) to the upper hinge (the 75th percentile) and therefore contains the middle half of the scores in the distribution. The median is shown as a line across the box.

For each run, the solutions set of two algorithms are compared using the coverage metric. Fig. 4 depicts the coverage metrics of $C(MAFI, MOGA)$ and $C(MOGA, MAFI)$ from 30 runs. In solving the small problem *m3o10*, Fig. 4 shows that the performance of MAFI and MOGA are almost the same. For another small problem *m4o20*, the non-dominated solutions obtained by MAFI dominates 80% of the solutions obtained by MOGA in average, while the non-dominated solutions obtained by MOGA only dominates 60% of the non-dominated solutions obtained by MAFI in average. As the complexity of problems increases, Fig. 4 shows that 80%-90% of the non-dominated solutions obtained by MOGA are weakly dominated by the non-dominated solutions obtained by MAFI in solving the problems *m4o20*, *m5o100*, *m5o200*, *m10o100* and *m10o200*. On the contrast, the non-dominated solutions of MOGA dominate nearly 3-10% of the non-dominated solutions obtained by MAFI. Fig. 5 shows the non-dominated solutions obtained by thirty runs of MAFI and MOGA in solving the *m10o200* problem. The results indicate that MAFI can converge to better solutions more quickly than MOGA. It reveals that the proposed schemata-guided local search strategy and fitness inheritance plays an important role in obtaining good solutions and accelerating the convergence speed.

TABLE I
THE PARAMETER SETTINGS OF MAFI AND MOGA.

Parameters	MAFI	MOGA
N_{pop}	115	100
E_{max}	115	100
p_s	0.25	0.25
p_i	0.5	N/A
p_c	0.6	0.6
p_m	0.05	0.05
$MaxLS$	3	N/A

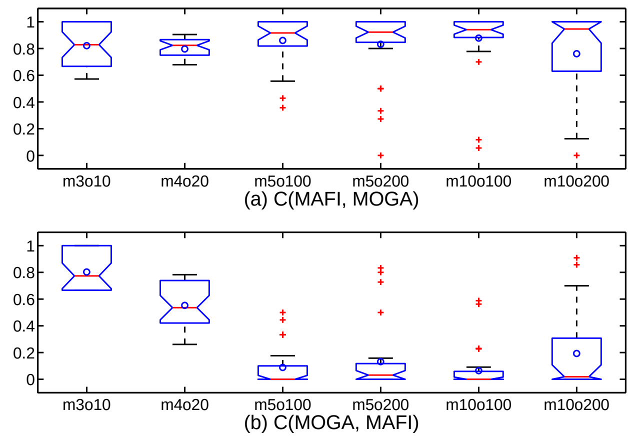

Fig. 4. Box plots based on the cover metric. (a) C(MAFI, MOGA), (b) C(MOGA, MAFI).

6 Conclusion

In this paper, a novel approach to solve flexible process sequencing problems using an multi-objective memetic algorithm MAFI is proposed. A schemata-guided local search

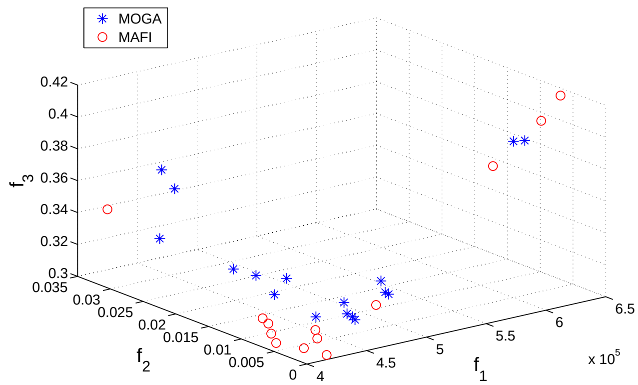

Fig. 5. The non-dominated solutions obtained by MAFI and MOGA in solving the *m10o200* problem, merged from 30 runs.

strategy and fitness inheritance are integrated in the proposed algorithm for enhancing the performance. Experimental results demonstrated that the quality of non-dominated solutions obtained by MAFI is better than that of MOGA in terms of convergence speed and accuracy using the same number of function evaluations. The results indicate that the proposed approach is an efficient approach to solving FPSs.

In addition, the advantages of the proposed approach are that MAFI can optimized multiple objectives without decomposing problems into sub-problems or using relative preferences of multiple objectives. While prior domain knowledge for the decomposition of problems or relative preferences of multiple objectives are not available, the proposed approach is an expedient method to solve flexible process sequencing problems. Moreover, the proposed approach can obtain a set of non-dominated solutions for decision makers in a single run. Decision makers can easily distinguish between the costs of different process sequences and choose more than one satisfactory process sequences at a time.

7 Acknowledgment

This work was supported by the National Science Council of Taiwan, R.O.C. under Contract NSC-96-2221-E-216-037-MY2 and NSC-095-SAF-I-564-616-TMS, and Chung-Hua University under Contract CHU-96-2221-E-216-037-MY2.

8 References

[1] H.-P. Wang, “A layered architecture for manufacturing operation planning,” *Comput. Ind. Eng.*, vol. 14, no. 2, pp. 201–210, 1988.

[2] A. Kusiak and G. Finke, “Selection of process plans in automated manufacturing systems,” *IEEE Transactions on Robotics and Automation*, vol. 4, no. 4, pp. 397–402, Aug 1988.

[3] C. Moon, Y.-Z. Li, and M. Gen, “Evolutionary algorithm for flexible process sequencing with multiple objectives,” in *Proceeding of IEEE International Conference on Computational Intelligence*, 1998, pp. 27–32.

[4] P. Brandimarte, “Exploiting process plan flexibility in production scheduling: A multi-objective approach,” *European Journal of Operational Research*, no. 114, pp. 59–71, 1999.

[5] H. C. Zhang and S. H. Huang, “A fuzzy approach to process plan selection,” *International Journal of Production Research*, vol. 32, no. 6, pp. 1265 – 1279, 1994.

[6] M. Gen and R. Cheng, *Genetic algorithms and engineering design*. New York: John Wiley, 1997, 1944- Mitsuo Gen, Runwei Cheng. ill. ; 24 cm.

[7] K. Deb, *Multi-objective optimization using evolutionary algorithms*, ser. Wiley-Interscience series in systems and optimization. John Wiley & Sons, 2001.

[8] R. Dawkins, *The selfish gene*. New York: Oxford University Press, 1976, 1941- Richard Dawkins 23 cm Includes index.

[9] H. Ishibuchi, T. Yoshida, and T. Murata, “Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 204–223, April 2003.

[10] J.-H. Chen and S.-Y. Ho, “A novel approach to production planning of flexible manufacturing systems using an efficient multi-objective genetic algorithm,” *International Journal of Machine Tools and Manufacture*, vol. 45, no. 7-8, pp. 949–957, 2005.

[11] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Pub. Co., 1989.

[12] J.-H. Chen, D. E. Goldberg, S.-Y. Ho, and K. Sastry, “Fitness inheritance in multi-objective optimization,” in *GECCO ’02: Proceedings of the Genetic and Evolutionary Computation Conference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 319–326.

[13] S.-Y. Ho, L.-S. Shu, and J.-H. Chen, “Intelligent evolutionary algorithms for large parameter optimization problems,” *IEEE Transaction on Evolutionary Computation*, vol. 8, no. 6, pp. 522–541, Dec. 2004.

[14] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: A comparative case study and the strengthen Pareto approach,” *IEEE Transaction on Evolutionary Computation*, vol. 4, no. 3, pp. 257–271, 1999.

Multi-objective Evolutionary Optimization of 3D Differentiated Sensor Network Deployment

Chih-Wei Kang and Jian-Hung Chen*
Department of Computer Science and Information Engineering
Chung-Hua University, Hsin-Chu 300, Taiwan

jh.chen@ieee.org*

ABSTRACT
This paper describes a multi-objective evolutionary approach for solving multi-objective 3D deployment problems in differentiated wireless sensor networks (WSNs). WSN is a wireless network consisting of spatially distributed autonomous sensors to monitor physical or environmental conditions. Deciding the location of sensor to be deployed on a terrain with the consideration different criteria is an important issue for the design of wireless sensor network. A multi-objective genetic algorithm is proposed to solve 3D differentiated WSN deployment problems with the objectives of the coverage of sensors, satisfaction of detection thresholds, and energy conservation. The preliminary experimental results demonstrated that the proposed approach is suitable for solving 3D deployment problems of WSNs with different requirements.

Categories and Subject Descriptors
J.6 [COMPUTER-AIDED ENGINEERING]: Computer-aided design (CAD)

General Terms
Algorithms, Design, Performance

Keywords
Wireless sensor network, multi-objective optimization, genetic algorithms

1. INTRODUCTION
A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous sensors to monitor physical or environmental conditions. WSN constitute a large number of applications related to national security, surveillance, military, health care, and home automation. Sensor nodes of a WSN are deployed over a region to sense events on geographical areas and transmit collected data to a sink node for further operations. Depending on the requirements, sensors could be deployed in diverse scenarios [6,9]. Therefore, deciding the location of sensor to be deployed on a terrain is an important issue. Several different objectives should be considered and fulfilled in the design phase of WSNs, such as the coverage and accuracy, reaction time and survivability of the sensor network. However, these objectives may be in conflict with one another and of different importance to mission planners [8].

Coverage is one of the fundamental issue in the deployment of WSNs. WSNs need to maintain sufficient coverage quality to capture the timely changing targets [10]. For enhanced coverage, a large number of sensors are typically deployed in the sensor field and, if the coverage areas of multiple sensors overlap, they may all report a target in their respective zones [5].

Differentiated sensor network deployment, which considers the satisfaction of detection levels in different geographical characteristics, is also an important issue [7]. In many real-world WSN applications, the supervised area can request different detection levels, depending on the event's location. Therefore, the sensing requirements are not uniformly distributed within the area. In other words, all the points of the area under monitoring are considered with the different importance. As a result, the deployment strategy of WSN should take into consideration the geographical characteristics of the monitored events.

Energy conservation for the lifetime of sensors is another rising issue [1]. Due to the limited energy resource in each sensor node, we need to utilize the sensors in an efficient manner so as to increase the lifetime of the network. There are two different approaches to the problem of conserving energy in sensor networks. The first approach is to plan a schedule of active sensors that enables other sensors to go into a sleep mode. The second approach is adjusting the sensing range of sensors for energy conservation. In this paper, we focus on adjusting the sensing range of each sensor in order to reduce the overlaps among sensing ranges while keep the detection ability above a predefined detection level.

In this paper, a 3D differentiated WSN deployment considering coverage, satisfaction of detection levels, and energy conservation is formulated into a multi-objective optimization problem. We represent the sensor field as a three-dimensional grid of points. Three objectives are to be optimized: maximizing coverage of sensors, satisfying the required probability of detection threshold, and minimizing the detection power by adjustable sensing range. To solve the aforementioned multi-objective optimization problem, we developed a multi-objective genetic algorithm (MOGA) framework. The proposed approach can obtain a set of non-dominated solutions. As a result, mission planner can deploy sensor nodes considering different requirements of applications.

2. RELATED WORK
2.1 WSN Deployment Problem
Coverage issue is one of the most important tasks in WSN. The ultimate goal is to have each location in the physical space of interest within the sensing range of at least one sensor. However, due to the number of sensors is limited, complete coverage cannot be guaranteed. Therefore, many approaches are proposed to deal with the 2D coverage problem. Oh et al. [8] proposed a genetic algorithm for the optimal selection of the number and type of

GECCO'09, July 8–12, 2009, Montréal, Québec, Canada.
ACM 978-1-60558-505-5/09/07.

sensors available from a suite of sensors. Dhawan et al. [3] proposed a novel searching algorithm based on improved NSGA-II to select an optimal cover set. It maintains the full coverage in large sensor networks by a small number of sensor nodes. For a practical approach, a probabilistic sensor detection model is adopted in combination with the detection error range and coverage threshold. Recently, Oktug et al. [9] proposed an approach to solve coverage problem by simulating sensor deployment strategies on a 3D terrain model and to find answers to questions that how many sensors are needed to cover a specified 3D terrain at a specified coverage percentage.

In recent years, how to efficiently utilize limited energy in a wireless sensor network has become an important issue. In [2], the problem is to prolong maximum network lifetime when all grid points are covered and sensor energy resources are constrained. In [6], the method used to extend the network lifetime is to divide the sensors into a number of sets, such that only one set is responsible for monitoring the targets, and all other sensors are in sleep mode. In the sleep mode, it consumes the least energy. If all the sensor nodes operate in the active mode simultaneously, an excessive amount of energy will be wasted and the data collected will be redundant. In [4], two new energy-efficient models of different sensing ranges are proposed. They used scheduling models with adjustable sensing ranges of each sensor in order to reduce the overlaps among detection ranges.

Different applications require different degrees of sensing coverage. While some applications may require a complete coverage in a region, others may only need a high percentage of coverage. Such WSN is called differentiated WSN [7]. In [11], three density control protocols by considering the tradeoff between energy usage and coverage was developed to select sensors. Few studies have considered the case of geographical irregularity of the sensed event. Aitsaadi et al. [7] presented a required minimum probability detection threshold of each grid point. They proposed a probabilistic event detection model and use a Tabu Search method to solve the differentiated WSN deployment problem.

2.2 Multi-objective Evolutionary Optimization

Assume the multi-objective functions are to be minimized. Mathematically, MOOPs can be represented as the following vector mathematical programming problems:

$$Minimize \quad F(Y) = \{F_1(Y), F_2(Y), ..., F_i(Y)\}. \qquad (1)$$

where Y denotes a solution and $F_i(Y)$ is generally a nonlinear objective function. Pareto dominance relationship and some related terminologies are introduced below. When the following inequalities hold between two solutions Y_1 and Y_2, Y_2 is a non-dominated solution and is said to dominate Y_1 ($Y_2 \succ Y_1$):

$$\forall i : F_i(Y_1) \geq F_i(Y_2) \wedge \exists j : F_j(Y_1) > F_j(Y_2). \qquad (2)$$

When the following inequality hold between two solutions Y_1 and Y_2, Y_2 is said to weakly dominate Y_1 ($Y_2 \succeq Y_1$):

$$\forall i : F_i(Y_1) \geq F_i(Y_2). \qquad (3)$$

A feasible solution Y * is said to be a Pareto-optimal solution if and only if there does not exist a feasible solution Y where Y dominates Y *, and the corresponding vector of Pareto-optimal solutions is called Pareto-optimal front.

By making use of Pareto dominance relationship, multi-objective evolutionary algorithms (MOEAs) are capable of performing the fitness assignment of multiple objectives without using relative preferences of multiple objectives. Thus, all the objective functions can be optimized simultaneously. As a result, MOEA seems to be an alternative approach to solving production planning and inspection planning problems on the assumption that no prior domain knowledge is available.

3. PROBLEM STATEMENT

3.1 Notations

In order to formulate problems, the following notations are introduced:

- i : sensor index, $i = 1,2,3,...,N$.
- j : grid point index, $j = 1,2,3,...,M$.
- k : sensing range index, $k = 1,2,3,...,K$.

3.2 Environment

We assume that N sensors $s_1, s_2, ..., s_N$ are deployed to cover the sensor field. Let the sensor field T consist of n_x, n_y, and n_z grid points $p_1, p_2, ..., p_M$ in the x, y, and z dimensions, respectively [5]. Each sensor has an initial sensor energy E and has the capability to adjust its sensor range. Sensing range options are $r_1, r_2, ..., r_K$, corresponding to energy consumptions of $e_1, e_2, ..., e_K$ and detection error ranges $f_1, f_2, ..., f_K$ ($f_k < r_k$) [6]. We assume that each grid point p_j in sensor field is associated a required minimum probability detection threshold, denoted $t(p_j)$.

3.3 Mathematical Formation of 3D Deployment Problem

3.3.1 Coverage

In many WSN applications, the main task is the surveillance of certain geographical areas [9]. Target location can be simplified considerably if the sensors are placed in such a way that every grid point in the sensor field is covered by sensors. In this way, the sensors reporting a target at time t uniquely identifies the grid location for the target at time t. The trajectory of a moving target can also be easily determined in this fashion from time series data [5].

Assume that sensor s_i is deployed at grid point. For any grid point p_j, the Euclidean distance between sensor s_i and grid point p_j is denoted as

$$d(s_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \qquad (4)$$

where x_i, x_j, y_i, y_j, z_i and z_j are coordinate location values. The way to compute the sensor and target coverage relationship is to consider that a sensor covers a target if the Euclidean distance between the sensor and target is no greater than a predefined sensing range. The following equation shows a binary detection model expressing the coverage $c_b(s_i, p_j)$ of a grid point p_j by sensor s_i.

$$c_b(s_i,p_j) = \begin{cases} 1, & if \ d(s_i,p_j) < r_k(s_i) \\ 0, & otherwise \end{cases} \qquad (5)$$

, where $r_k(s_i)$ is the sensing range of the sensor s_i. In this paper, we use this binary detection model in coverage problem. Thus, the coverage rate optimization problem F_1 can be defined by

$$Max.\quad F_1 = \frac{\sum_{j=1}^{M} c_b(p_j)}{M} \tag{6}$$

, where $c_b(p_j)$ is the coverage of all sensors at grid point p_j by the Equation (5). This objective is to be maximized.

3.3.2 Detection Probability Thresholds

We suppose that the sensor field is characterized by the geographical irregularity of the sensed events. This assumption is justified by many realistic WSN applications case studies. To efficiently monitor the area, and since we consider a probabilistic detection model, we assume that, to each grid point p_j in sensor field is associated a required minimum probability detection threshold, denoted $t(p_j)$. Some grid points p_j in sensor field T will have a low detection probability if they are covered only by one sensor and far from other sensors. In this case, it is necessary to make the detection area overlapped to compensate for the low detection probability of the grid points that are far from any sensor. Ideally, a good WSN deployment algorithm should lead to obtain that each p_j in T the measured detection probability of that point is greater than $t(p_j)$ [7].

In reality, binary detection model has limitations due to the imprecise detection probability, which plays a significant role in sensor detection [3]. Hence, a detection error range is introduced to measure the uncertainty of sensor detection [3]. More precisely, we assume that event detection ability of a sensor diminishes as its distance to the sensed point increases [7]. A probabilistic detection model is expressed as

$$c_p(s_i,p_j) = \begin{cases} 0, & if\ r_k(s_i) + f_k(s_i) \le d(s_i,p_j) \\ e^{-\lambda\alpha^{\beta}}, & if\ r_k(s_i) - f_k(s_i) < d(s_i,p_j) < r_k(s_i) + f_k(s_i) \\ 1, & if\ r_k(s_i) - f_k(s_i) \ge d(s_i,p_j) \end{cases} \tag{7}$$

, where $\alpha = d(s_i, p_j) - (r_k(s_i) - f_k(s_i))$, λ and β are parameters that measure the detection probabilities when an object is within a certain distance from the sensor, and $f_k(s_i)$ is the error ranges of the sensor s_i. Each sensor s_i has a detection probability $c_p(s_i, p_j)$ at grid point p_j. A grid point p_j might be covered by more than one detection range of different sensors [2]. When a detection area is overlapped by multiple sensors, the closer are the sensors to each other, the higher is the detection probability of the grid points [3]. The conjunctive detection probability of all sensors at grid point p_j is given by

$$c_p(p_j) = 1 - \prod_{i=1}^{N}(1 - c_p(s_i, p_j)). \tag{8}$$

The optimization of the satisfaction required probability of detection threshold F_2 is expressed by:

$$Max.\quad F_2 = \frac{\sum_{j=1}^{M} DP(p_j)}{\sum_{j=1}^{M} t(p_j)} \tag{9}$$

,where $DP(p_j) = \begin{cases} t(p_j) & if\ c_p(p_j) - t(p_j) \ge 0 \\ 0 & otherwise \end{cases}$.

This objective is to be maximized.

3.3.3 Energy Consumption

In terms of energy consumption, we only consider the energy used in sensing, not including the power consumed by radio communication and computation. The sensing ranges of a sensor determine the energy consumed by the sensor [6]. We attempt to make the detection regions of sensors not overlapped, thereby minimizing the wasted overlap area and covering more grid points with a small number of sensors. We apply a energy model in our evaluation, in which the power consumption is proportional to the square of the sensing range r_k [11]. The energy consumption model is expressed as follows:

$$e_k(s_i) = \mu \times r_k(s_i)^2 \tag{10}$$

, where μ is an energy consumption parameter. The optimization of the detection power minimization with adjustable sensing range F_3 can be formulated as

$$Min.\quad F_3 = \frac{\sum_{i=1}^{N} e_k(s_i)}{\sum_{i=1}^{N} e_{\max}(s_i)} \tag{11}$$

, where $e_{max}(s_i)$ is the maximum detection range of each sensor. This objective is to be minimized.

4. MULTI-OBJECTIVE GENETIC ALGORITHM

4.1 Chromosome Representation

A chromosome has gene information for solving the problem in MOGA. Each chromosome has fixed gene size, which is determined by the number of sensors in the WSN. Each gene has a *x, y,* and *z* coordinate location and a sensing range. The ranges of each gene of coordinate location are [0, n_x], [0, n_y], and [0, n_z] in the *x, y,* and *z* dimensions. Hence these sensors will have coordinate values to denote their location. Each gene of sensing range is one of $r_1, r_2, \ldots, r_K$, which represent the detection ability of the sensor.

4.2 Fitness Assignment

We use a generalized Pareto-based scale-independent fitness function (GPSIFF) considering the quantitative fitness values in Pareto space for both dominated and non-dominated individuals. GPSIFF makes the best use of Pareto dominance relationship to evaluate individuals using a single measure of performance. The used GPSIFF is briefly described below. Let the fitness value of an individual X be a tournament-like score obtained from all participant individuals by the following function:

$$F(X) = p - q + c \tag{12}$$

, where p is the number of individuals which can be dominated by the individual X, and q is the number of individuals which can dominate the individual X in the objective space. Generally, a constant c can be optionally added in the fitness function to make fitness values positive. c is usually set to the number of all participant individuals.

4.3 Genetic Operators

The genetic operators used in the proposed approach are widely used in literature. The selection operator uses a binary tournament selection without replacement, which works as follows. Choose two individuals randomly from the population and copy the better individual into the intermediate population.

Crossover is a recombination process in which genes from two selected parents are recombined to generate offspring chromosomes. The uniform crossover is used in MOGA. In a uniform crossover operation, first requires a randomly created binary string, called crossover mask. The genes of offspring chromosomes are swapped from the parents according to this mask. If the crossover mask bit is 0, then the characters in the corresponding string position are not swapped and if the crossover mask bit is 1, than the mating string characters at that position are swapped.

A simple mutation operator is used to alter genes. For each gene, randomly generate a real value from the range [0, 1]. If the value is smaller than the mutation probability p_m, replace its index with a randomly generated integer among its possible values.

4.4 Procedure of MOGA

The procedure of MOGA is written as follows:

Input: population size N_{pop}, recombination probability p_c, mutation probability p_m, the number of maximum generations G_{max}.

Output: The optimum solutions ever found in P.

Step 1: Initialization Randomly generate an initial population P of N_{pop} individuals.

Step 2: Evaluation For each individual in the population, compute all objective function values F_1, F_2, and F_3.

Step 3: Fitness assignment Assign each individual a fitness value by using GPSIFF.

Step 4: Selection Select N_{pop} individuals from the population to form a new population using the binary tournament selection.

Step 5: Recombination Perform the uniform crossover operation with a recombination probability p_c.

Step 6: Mutation Apply the mutation operator to each gene in the individuals with a mutation probability p_m.

Step 7: Termination test If a stopping condition is satisfied, stop the algorithm. Otherwise, go to Step 2.

5. RESULT AND DISCUSSION

In this section, we present some results of simulation experiments as the performance evaluation of our proposed algorithm.

5.1 Simulation Environment and Parameters

A 3D WSN deployment benchmark generator for WSN environment is designed to generate different scale of sensor fields with different models of detection probability thresholds.

In this paper, a sensor field with 50×50×50 grid points is used. The same terrain with four different required minimum detection probability thresholds are illustrated as four different benchmarks. The detection probability thresholds considered in this paper are decreasing linear, normal, Poisson, and exponential distributions, respectively. Figure 1 illustrates a terrain with linear decreasing thresholds. For the sensors of WSN, we assume each sensor has five adjustable sensing ranges 6, 8, 10, 12, 14, and the detection error ranges are half of the sensing range of each sensor. The power consumption parameter μ is 1. The probabilistic detection model parameter β is 0.5 and the detection radio wave parameter λ is 0.5.

The parameter settings of MOGA are listed as follows: population size N_{pop}=200, recombination probability p_c=0.9, mutation probability p_m=0.01, the number of maximum generations G_{max}=500. The number of sensor nodes to be deployed is 20. Thirty independent runs are conducted.

Figure 2-3 depicts the box plots of obtained non-dominated solutions and the maximum and minimum objective values obtained in different objective functions. Figure 4-6 depicts the convergence speed of a typical run in solving the 3D WSN deployment problem with four different required minimum detection probability thresholds. The results indicate that different detection probability thresholds pose different difficulties for MOGA. The problems with normal and Poisson distributions are more difficult to find a good deployment plan than problems with decreasing linear and exponential distributions.

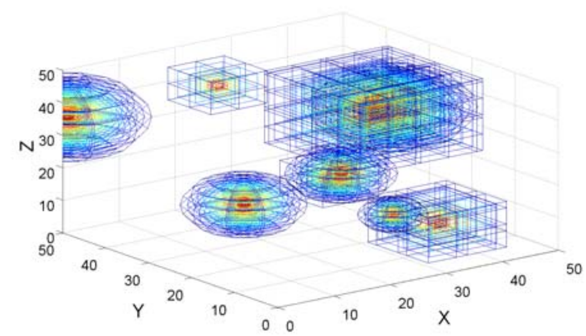

Figure 1. A terrain with decreasing linear detection probability thresholds.

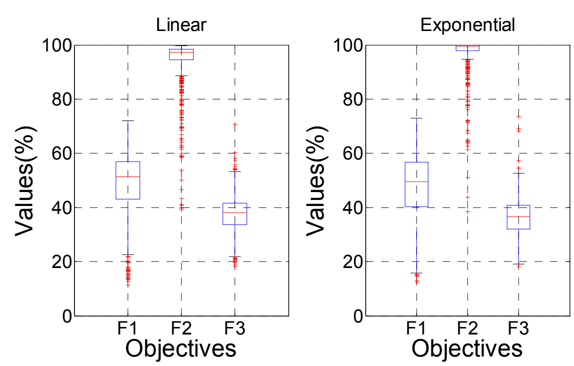

Figure 2. Box plots of non-dominated solutions for solving the 3D deployment problem with linear and exponential distributions probability thresholds.

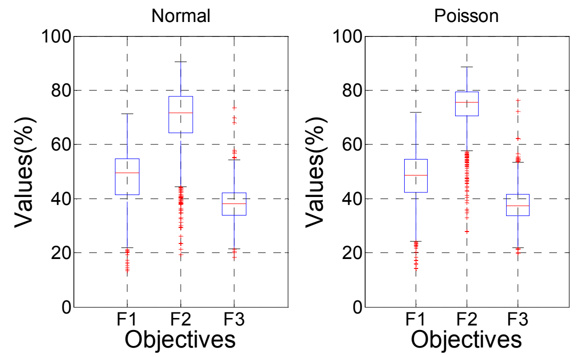

Figure 3. Box plots of non-dominated solutions for solving the 3D deployment problem with normal and Poisson distributions probability thresholds.

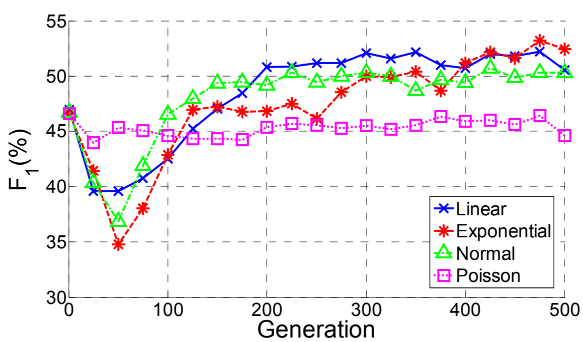

Figure 4. The mean objective value F_1 of non-dominated solutions in each generation, for four problems with different required detection probability thresholds.

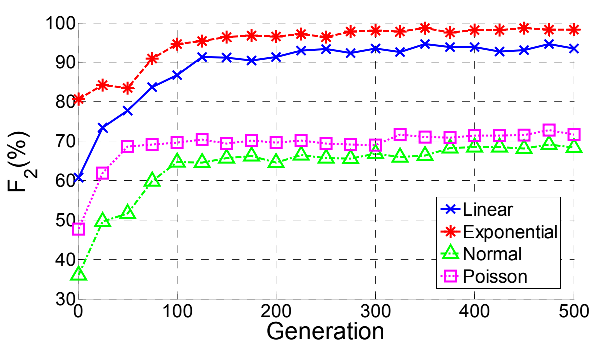

Figure 5. The mean objective value F_2 of non-dominated solutions in each generation, for four problems with different required detection probability thresholds.

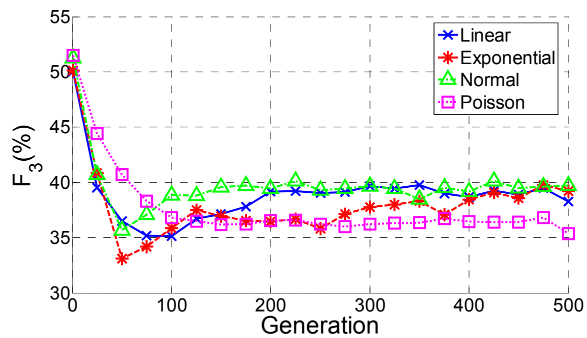

Figure 6. The mean objective value F_3 of non-dominated solutions in each generation, for four problems with different required detection probability thresholds.

6. CONCLUSION

In this paper, a multi-objective evolutionary approach is proposed to solve 3D differentiated WSN deployment problems. Experimental results demonstrated MOGA is capable of optimizing coverage, satisfaction of detection levels, and energy conservation and provide mission planers a set of non-dominated solutions for deployment of sensor nodes. The results also indicates that some problems with unusual detection probability thresholds requirements may require more computation time or different techniques for MOGA than those of problems with usual detection probability thresholds requirements.

7. ACKNOWLEDGMENTS

This work was supported by the National Science Council of Taiwan, R.O.C. under Contract NSC-96-2221-E-216-037-MY2, and Chung-Hua University under Contract CHU-96-2221-E-216-037-MY2.

8. REFERENCES

[1] A. Dhawan, C. T. Vu, A. Zelikovsky, Y. Li, and S. K. Prasad, “Maximum Lifetime of Sensor Networks with Adjustable Sensing Range,” *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006. Seventh ACIS International Conference on*, pp. 285-289.

[2] H.-C. Jang and H.-C. Lee, “Efficient energy management to prolong wireless sensor network lifetime,” *Internet, 2007. ICI 2007. 3rd IEEE/IFIP International Conference in Central Asia on*, pp. 1-4.

[3] J. Jia, J. Chen, G.-R. Chang, and Y.-Y. Wen, “Efficient Cover Set Selection in Wireless Sensor Networks,” *Acta Automatica Sinica*, vol. 34, no. 9, 2008, pp. 1157-1162.

[4] J. Wu and S. Yang, “Coverage issue in sensor networks with adjustable ranges,” *Parallel Processing Workshops, 2004. ICPP 2004 Workshops. Proceedings. 2004 International Conference on*, pp. 61-68.

[5] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, “Grid coverage for surveillance and target location in distributed sensor networks,” *Computers, IEEE Transactions on*, vol. 51, no. 12, 2002, pp. 1448-1453.

[6] M. Cardei, J. Wu, and M. Lu, “Improving network lifetime using sensors with adjustable sensing ranges,” *Int. J. Sen. Netw.*, vol. 1, no. 1/2, 2006, pp. 41-49.

[7] N. Aitsaadi, N. Achirt, K. Boussetta, and G. Pujolle, “A Tabu Search Approach for Differentiated Sensor Network Deployment,” *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pp. 163-167.

[8] S. C. Oh, C. H. Tan, F. W. Kong, Y. S. Tan, K. H. Ng, G. W. Ng, and K. Tai, “Multiobjective optimization of sensor network deployment by a genetic algorithm,” *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pp. 3917-3921.

[9] S. Oktug, A. Khalilov, and H. Tezcan, “3D Coverage Analysis under Heterogeneous Deployment Strategies in Wireless Sensor Networks,” *Telecommunications, 2008. AICT '08. Fourth Advanced International Conference on*, pp. 199-204.

[10] S. Zhang, Y. Liu, J. Pu, X. Zeng, and Z. Xiong, “An Enhanced Coverage Control Protocol for Wireless Sensor Networks,” *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on*, pp. 1-7.

[11] Y. Shang and H. Shi, “Coverage and energy tradeoff in density control on sensor networks,” *Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on*, pp. 564-570 Vol. 561.

2009 International Conference on Computational Science and Engineering

An Evolutionary Approach for Multi-objective 3D Differentiated Sensor Network Deployment

Chih-Wei Kang and Jian-Hung Chen*
Department of Computer Science and Information Engineering
Chung-Hua University, Hsin-Chu 300, Taiwan
jh.chen@ieee.org*

***Abstract*—This paper describes a multi-objective evolutionary approach for solving multi-objective 3D deployment problems in differentiated wireless sensor networks (WSNs). WSN is a wireless network consisting of spatially distributed autonomous sensors to monitor physical or environmental conditions. Deciding the location of sensor to be deployed on a terrain with the consideration of different criteria is an important issue for the design of wireless sensor network. A multi-objective genetic algorithm is proposed to solve 3D differentiated WSN deployment problems with the objectives of the coverage of sensors, satisfaction of detection levels, and energy conservation. The preliminary experimental results demonstrated that the proposed approach is suitable for solving 3D deployment problems of WSNs with different requirements.**

Keywords- Wireless sensor network, multi-objective optimization, genetic algorithms

I. Introduction

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous sensors to monitor physical or environmental conditions. WSN constitute a large number of applications related to national security, surveillance, military, health care, and home automation. Sensor nodes of a WSN are deployed over a region to sense events on geographical areas and transmit collected data to a sink node for further operations. Depending on the requirements, sensors could be deployed in diverse scenarios [4,9]. Therefore, deciding the location of sensor to be deployed on a terrain is an important issue. Several different objectives should be considered and fulfilled in the design phase of WSNs, such as the coverage and accuracy, reaction time and survivability of the sensor network. However, these objectives may be in conflict with one another and of different importance to mission planners [10].

Coverage is one of the fundamental issue in the deployment of WSNs. WSNs need to maintain sufficient coverage quality to capture the timely changing targets [13]. For enhanced coverage, a large number of sensors are typically deployed in the sensor field and, if the coverage areas of multiple sensors overlap, they may all report a target in their respective zones [3].

Differentiated sensor network deployment, which considers the satisfaction of detection levels in different geographical characteristics, is also an important issue [1]. In many real-world WSN applications, such as underwater sensor deployment, the supervised area may require different detection levels, depending on the event's location. Therefore, the sensing requirements are not uniformly distributed within the area. In other words, all the points of the area under monitoring are considered with the different importance. As a result, the deployment strategy of WSN should take into consideration the geographical characteristics of the monitored events.

Energy conservation for the lifetime of sensors is another rising issue [5]. Due to the limited energy resource in each sensor node, we need to utilize the sensors in an efficient manner so as to increase the lifetime of the network. There are two different approaches to the problem of conserving energy in sensor networks. The first approach is to plan a schedule of active sensors that enables other sensors to go into a sleep mode. The second approach is adjusting the sensing range of sensors for energy conservation. In this paper, we focus on adjusting the sensing range of each sensor in order to reduce the overlaps among sensing ranges while keep the detection ability above a predefined detection level.

In this paper, a 3D differentiated WSN deployment considering coverage, satisfaction of detection levels, and energy conservation is formulated into a multi-objective optimization problem. We represent the sensor field as a three-dimensional grid of points. Three objectives are to be optimized: maximizing coverage of sensors, satisfying the required probability of detection level, and minimizing the detection power by adjustable sensing range. To solve the aforementioned multi-objective optimization problem, we developed a multi-objective genetic algorithm (MOGA) framework. The proposed approach can obtain a set of non-dominated solutions for mission planner to deploy sensor nodes considering different requirements of applications.

II. Related Work

A. *WSN Deployment Problem*

Coverage issue is one of the most important tasks in WSN. The ultimate goal is to have each location in the physical space of interest within the sensing range of at least one sensor. However, due to the number of sensors is limited, complete coverage cannot be guaranteed. Therefore, many approaches are proposed to deal with the 2D coverage problem. Oh et al.

978-0-7695-3823-5/09 $26.00 © 2009 IEEE
DOI 10.1109/CSE.2009.329

[10] proposed a genetic algorithm for the optimal selection of the number and type of sensors available from a suite of sensors. Dhawan et al. [7] proposed a novel searching algorithm based on improved NSGA-II to select an optimal cover set. It maintains the full coverage in large sensor networks by a small number of sensor nodes. For a practical approach, a probabilistic sensor detection model is adopted in combination with the detection error range and coverage threshold. Recently, Oktug et al. [9] proposed an approach to solve coverage problem by simulating sensor deployment strategies on a 3D terrain model and to find answers to questions that how many sensors are needed to cover a specified 3D terrain at a specified coverage percentage.

Different applications require different degrees of sensing coverage. While some applications may require a complete coverage in a region, others may only need a high percentage of coverage. Such WSN is called differentiated WSN [1]. Take underwater sensor deployment [2] as an example, sensor field of underwater is characterized by the geographical irregularity of the sensed events because some area may be inaccessible or the event area may not be uniformly distributed. To efficiently monitor such area with differentiated detection levels, fulfillment of detection levels in different area is the major concerns instead of maximizing the coverage of sensors. In [11], three density control protocols by considering the tradeoff between energy usage and coverage was developed to select sensors. Few studies have considered the case of geographical irregularity of the sensed event. Aitsaadi et al. [1] proposed a probabilistic event detection model. In this model, each grid point has a required minimum probability detection threshold. A tabu Search method is proposed to solve this differentiated WSN deployment problem.

In recent years, utilizing limited energy efficiently in a wireless sensor network has become an important issue. In [8], the problem is to prolong maximum network lifetime when all grid points are covered and sensor energy resources are constrained. In [4], they proposed a method to extend the network lifetime is to divide the sensors into a number of sets, such that only one set is responsible for monitoring the targets, and all other sensors are in sleep mode. In the sleep mode, it consumes the least energy. If all the sensor nodes operate in the active mode simultaneously, an excessive amount of energy will be wasted and the data collected will be redundant. In [12], two new energy-efficient models of different sensing ranges are proposed. They used scheduling models with adjustable sensing ranges of each sensor in order to reduce the overlaps among detection ranges.

B. Multi-objective Evolutionary Optimization

Assume the multi-objective functions are to be minimized. Mathematically, MOOPs can be represented as the following vector mathematical programming problems

$$\textit{Minimize} \quad F(Y) = \{F_1(Y), F_2(Y), ..., F_i(Y)\}. \qquad (1)$$

where Y denotes a solution and $f_i(Y)$ is generally a nonlinear objective function. Pareto dominance relationship and some related terminologies are introduced below. When the following inequalities hold between two solutions Y_1 and Y_2, Y_2 is a non-dominated solution and is said to dominate Y_1 ($Y_2 \succ Y_1$):

$$\forall i : F_i(Y_1) > F_i(Y_2) \wedge \exists j : F_j(Y_1) > F_j(Y_2). \qquad (2)$$

When the following inequality hold between two solutions Y_1 and Y_2, Y_2 is said to weakly dominate Y_1 ($Y_2 \succeq Y_1$):

$$\forall i : F_i(Y_1) \geq F_i(Y_2). \qquad (3)$$

A feasible solution Y^* is said to be a Pareto-optimal solution if and only if there does not exist a feasible solution Y where Y dominates Y^*, and the corresponding vector of Pareto-optimal solutions is called Pareto-optimal front.

By making use of Pareto dominance relationship, multi-objective evolutionary algorithms (MOEAs) are capable of performing the fitness assignment of multiple objectives without using relative preferences of multiple objectives. Thus, all the objective functions can be optimized simultaneously. As a result, MOEA seems to be an alternative approach to solving production planning and inspection planning problems on the assumption that no prior domain knowledge is available [6].

III. PROBLEM STATEMENT

A. Notations

In order to formulate problems, the following notations are introduced:

- i : sensor index, $i = 1,2,3,...,N$.
- j : grid point index, $j = 1,2,3,...,M$.
- k : sensing range index, $k = 1,2,3,...,K$.

B. Environment

We assume that N sensors $s_1,s_2,..., s_N$ are deployed to cover the sensor field. Let the sensor field T consist of n_x, n_y, and n_z grid points $p_1,p_2,..., p_M$ in the x, y, and z dimensions, respectively [3]. Each sensor has an initial sensor energy E and has the capability to adjust its sensor range. Sensing range options are $r_1,r_2,..., r_K$, corresponding to energy consumptions of $e_1,e_2,..., e_K$ and detection error ranges $f_1,f_2,..., f_K$ ($f_k < r_k$) [4]. We assume that each grid point p_j in sensor field is associated a required minimum probability detection level, denoted $t(p_j)$.

C. Mathematical Formation of 3D Deployment Problem

1) Maximization of Coverage

In many WSN applications, the main task is the surveillance of certain geographical areas [9]. Target location can be simplified considerably if the sensors are placed in such a way that every grid point in the sensor field is covered by sensors. In this way, the sensors reporting a target at time t uniquely identifies the grid location for the target at time t. The trajectory of a moving target can also be easily determined in this fashion from time series data [3].

Assume that sensor s_i is deployed at grid point. For any grid point p_j, the Euclidean distance between sensor s_i and grid point p_j is denoted as

$$d(s_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (4)$$

where x_i, x_j, y_i, y_j, z_i and z_j are coordinate location values. The way to compute the sensor and target coverage relationship is to consider that a sensor covers a target if the Euclidean distance between the sensor and target is no greater than a predefined sensing range. The following equation shows a binary coverage model expressing the coverage $c_b(s_i, p_j)$ of a grid point p_j by sensor s_i.

$$c_b(s_i,p_j) = \begin{cases} 1, & \text{if } d(s_i,p_j) < r_k(s_i) \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

, where $r_k(s_i)$ is the sensing range of the sensor s_i.

The coverage rate optimization problem F_1 can be defined by

$$Max. \quad F_1 = \frac{\sum_{j=1}^{M} c_b(p_j)}{M} \quad (6)$$

, where $c_b(p_j)$ is the coverage of all sensors at grid point p_j by the Equation (5). This objective is to be maximized.

2) Maximization of Differentiated Detection Levels

Considering differentiated detection levels, assumed that each grid point p_j in sensor field T is associated a required minimum detection level, denoted $t(p_j)$. A terrain may have different required detection levels, as illustrated in Figure 1. Ideally, a good deployment for differentiated WSN should satisfy the following condition: for each p_j in T, the measured detection probability of p_j should be greater than or equal to $t(p_j)$ [1].

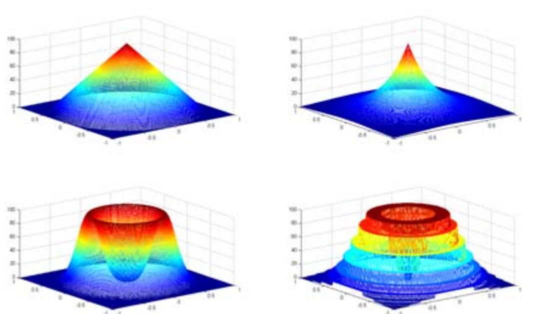

Figure 1. Terrain with different required detection levels: decreasing linear, normal, Poisson, and exponential distributions.

In literature, a 0/1 binary detection model for grid points is often used if a grid is covered by a sensor. However, in reality, the detection of events may be influence by weather or obstacles. In such cases, the 0/1 binary detection model has limitations due to the imprecise detection probability, which plays a significant role in sensor detection [7]. Hence, a detection error range is introduced to measure the uncertainty of sensor detection [7]. Each grid point covered by sensors has different detection probabilities according to their realistic conditions, such as distance to sensors or weather conditions. If a gird point in sensor field T is covered only by one sensor and far from other sensors, it may have a low detection probability. In this case, it is necessary to reallocate sensors, so that the detection area of sensors can be overlapped to compensate for the low detection probability of those grid points that are far from any sensor.

In this paper, we adopted a probabilistic detection model for sensor deployment [1]. Assume that event detection probability of a sensor diminishes as its distance to the sensed point increases. A probabilistic detection model of sensors is expressed as

$$c_p(s_i,p_j) = \begin{cases} 0, & \text{if } r_k(s_i) + f_k(s_i) \le d(s_i,p_j) \\ e^{-\lambda\alpha^{\beta}}, & \text{if } r_k(s_i) - f_k(s_i) < d(s_i,p_j) < r_k(s_i) + f_k(s_i) \\ 1, & \text{if } r_k(s_i) - f_k(s_i) \ge d(s_i,p_j) \end{cases} \quad (7)$$

, where $\alpha = d(s_i, p_j) - (r_k(s_i) - f_k(s_i))$, λ and β are parameters that measure the detection probabilities when an object is within a certain distance from the sensor, and $f_k(s_i)$ is the error ranges of the sensor s_i. Each sensor s_i has a detection probability $c_p(s_i, p_j)$ at grid point p_j. A grid point p_j might be covered by more than one detection range of different sensors [8]. When a detection area is overlapped by multiple sensors, the closer are the sensors to each other, the higher is the detection probability of the grid points [7]. The conjunctive detection probability of all sensors at grid point p_j is given by

$$c_p(p_j) = 1 - \prod_{i=1}^{N} (1 - c_p(s_i, p_j)). \quad (8)$$

The optimization of the satisfaction required probability of detection level F_2 is expressed by:

$$Max. \quad F_2 = \frac{\sum_{j=1}^{M} DP(p_j)}{\sum_{j=1}^{M} t(p_j)} \quad (9)$$

,where $DP(p_j) = \begin{cases} t(p_j) & \text{if } c_p(p_j) - t(p_j) \ge 0 \\ 0 & \text{otherwise} \end{cases}$.

This objective is to be maximized.

3) Minimization of Energy Consumption

In terms of energy consumption, we only consider the energy used in sensing, but not including the power consumed by radio communication and computation. The sensing ranges of a sensor determine the energy consumed by the sensor [4]. We attempt to make the detection regions of sensors not overlapped, thereby minimizing the wasted overlap area and covering more grid points with a small number of sensors. We apply an energy model in our evaluation, in which the power consumption is proportional to the square of the sensing range r_k [11]. The energy consumption model is expressed as follows:

$$e_k(s_i) = \mu \times r_k(s_i)^2, \quad (10)$$

where μ is an energy consumption parameter. The optimization of the detection power minimization with adjustable sensing range F_3 can be formulated as

$$Min. \quad F_3 = \frac{\sum_{i=1}^{N} e_k(s_i)}{\sum_{i=1}^{N} e_{\max}(s_i)} \tag{11}$$

, where $e_{max}(s_i)$ is the maximum detection range of each sensor. This objective is to be minimized.

IV. MULTI-OBJECTIVE GENETIC ALGORITHM

A. Chromosome Representation

A chromosome has gene information for solving the problem in MOGA. Each chromosome has fixed gene size, which is determined by the number of sensors in the WSN. Each gene has a *x, y,* and *z* coordinate location and a sensing range. The ranges of each gene of coordinate location are [0, n_x], [0, n_y], and [0, n_z] in the *x, y,* and *z* dimensions. Hence these sensors will have coordinate values to denote their location. Each gene of sensing range is one of $r_1, r_2, \ldots, r_K$, which represent the detection ability of the sensor.

B. Fitness Assignment

We use a generalized Pareto-based scale-independent fitness function (GPSIFF) considering the quantitative fitness values in Pareto space for both dominated and non-dominated individuals. GPSIFF makes the best use of Pareto dominance relationship to evaluate individuals using a single measure of performance. The used GPSIFF is briefly described below. Let the fitness value of an individual *Y* be a tournament-like score obtained from all participant individuals by the following function:

$$F(Y) = p - q + c \tag{12}$$

, where *p* is the number of individuals which can be dominated by the individual *Y*, and *q* is the number of individuals which can dominate the individual *Y* in the objective space. Generally, a constant *c* can be optionally added in the fitness function to make fitness values positive. *c* is usually set to the number of all participant individuals.

C. Genetic Operators

The genetic operators used in the proposed approach are widely used in literature. The selection operator uses a binary tournament selection without replacement, which works as follows. Choose two individuals randomly from the population and copy the better individual into the intermediate population.

Crossover is a recombination process in which genes from two selected parents are recombined to generate offspring chromosomes. The uniform crossover is used in MOGA. In a uniform crossover operation, first requires a randomly created binary string, called crossover mask. The genes of offspring chromosomes are swapped from the parents according to this mask. If the crossover mask bit is 0, then the characters in the corresponding string position are not swapped and if the crossover mask bit is 1, than the mating string characters at that position are swapped.

A simple mutation operator is used to alter genes. For each gene, randomly generate a real value from the range [0, 1]. If the value is smaller than the mutation probability pm, replace its index with a randomly generated integer among its possible values.

D. Procedure of MOGA

The procedure of MOGA is written as follows:

Input: population size N_{pop}, recombination probability p_c, mutation probability p_m, the number of maximum generations G_{max}.

Output: The optimum solutions ever found in *P*.

Step 1: Initialization Randomly generate an initial population *P* of N_{pop} individual*s*.

Step 2: Evaluation For each individual in the population, compute all objective function values F_1, F_2, and F_3.

Step 3: Fitness assignment Assign each individual a fitness value by using GPSIFF.

Step 4: Selection Select N_{pop} individuals from the population to form a new population using the binary tournament selection.

Step 5: Recombination Perform the uniform crossover operation with a recombination probability p_c.

Step 6: Mutation Apply the mutation operator to each gene in the individuals with a mutation probability p_m.

Step 7: Termination test If a stopping condition is satisfied, stop the algorithm. Otherwise, go to Step 2.

V. RESULT AND DISCUSSION

In this section, we present some results of simulation experiments as the performance evaluation of our proposed algorithm.

A. Simulation Environment and Parameters

A 3D WSN deployment benchmark generator for WSN environment is designed to generate different scale of sensor fields with different models of detection probability levels.

In this paper, a sensor field with 50×50×50 grid points is used. The same terrain with four different required minimum detection probability levels are illustrated as four different benchmarks. The detection probability levels considered in this paper are decreasing linear, normal, Poisson, and exponential distributions, respectively. Figure 2 illustrates a terrain with linear decreasing levels. For the sensors of WSN, we assume each sensor has five adjustable sensing ranges 6, 8, 10, 12, 14, and the detection error ranges are half of the sensing range of each sensor. The power consumption parameter μ is 1. The probabilistic detection model parameter β is 0.5 and the detection radio wave parameter λ is 0.5.

The parameter settings of MOGA are listed as follows: population size N_{pop}=200, recombination probability p_c=0.9, mutation probability p_m=0.01, the number of maximum generations G_{max}=500 and 1000. Thirty independent runs are conducted for each problem.

To identify the difficulties of problems and evaluate the performance of our algorithm, the number of sensor nodes to be deployed is limited to 20 and 50, respectively. Figures 3-7 show the results of deployment using 20 sensors. Figures 8-12 show the results of deployment using 50 sensors.

Figures 3,4,8,9 depict the box plots of obtained non-dominated solutions and the maximum and minimum objective values obtained in different objective functions, using 20 and 50 sensors. Figures 5-7 and 10-12 depict the convergence speed of a typical run in solving the 3D WSN deployment problem with four different required minimum detection probability levels, using 20 and 50 sensor nodes. The results indicate that different detection levels pose different difficulties for MOGA. The problems with normal and Poisson detection levels are more difficult to find a good deployment plan than problems with decreasing linear and exponential detection levels using the same number of sensors. The number of sensors required for a terrain with normal and Poisson detection levels should be bigger than the same terrain with decreasing linear and exponential detection levels.

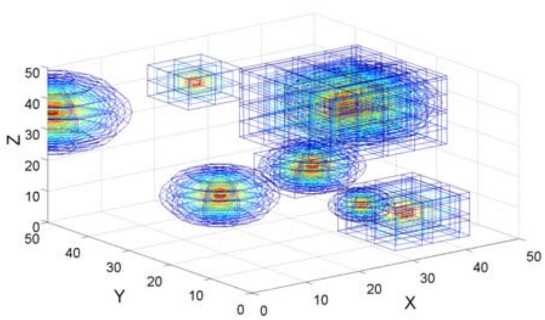

Figure 2. A terrain with decreasing linear detection levels.

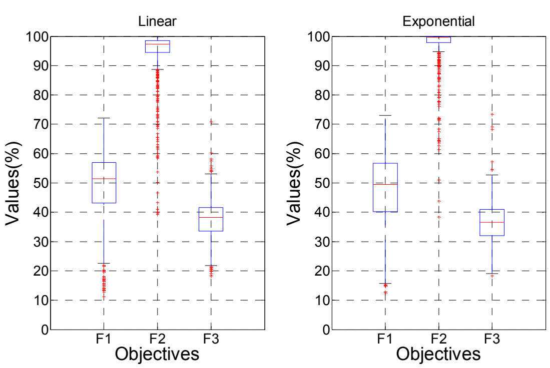

Figure 3. Box plots of non-dominated solutions for solving the 3D deployment problem with linear and exponential detection levels, using 20 sensors.

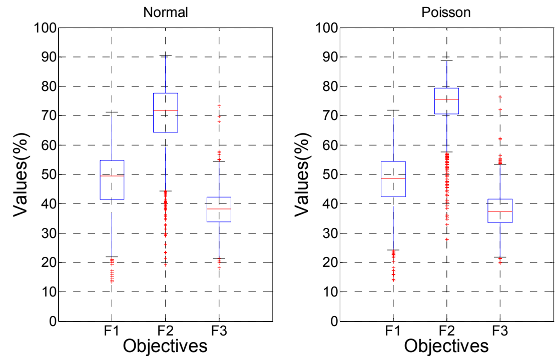

Figure 4. Box plots of non-dominated solutions for solving the 3D deployment problem with normal and Poisson detection levels, using 20 sensors.

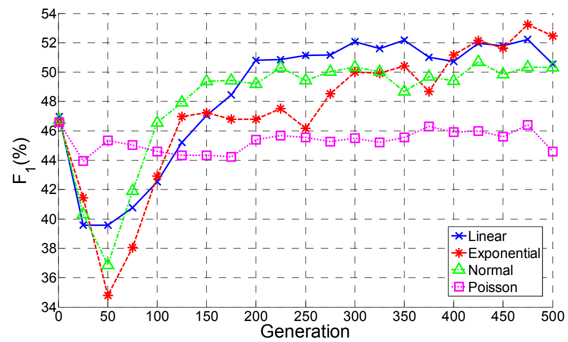

Figure 5. The mean objective value F_1 of non-dominated solutions in each generation, for four problems with different detection levels, using 20 sensors.

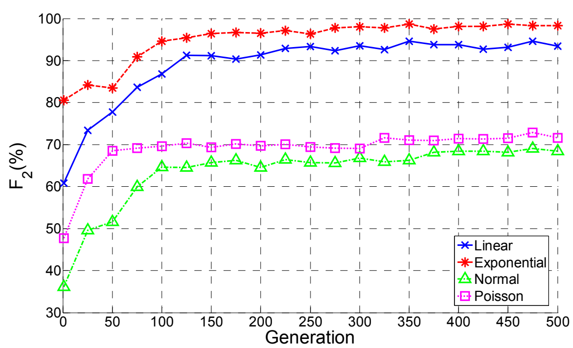

Figure 6. The mean objective value F_2 of non-dominated solutions in each generation, for four problems with different detection levels, using 20 sensors.

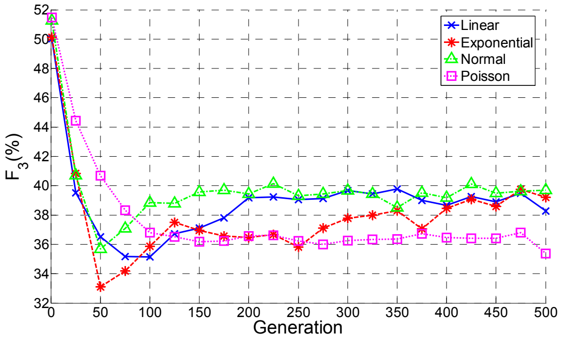

Figure 7. The mean objective value F_3 of non-dominated solutions in each generation, for four problems with different required detection levels, using 20 sensors.

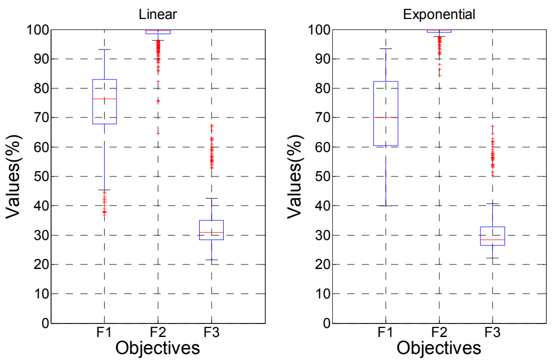

Figure 8. Box plots of non-dominated solutions for solving the 3D deployment problem with linear and exponential detection levels, using 50 sensors.

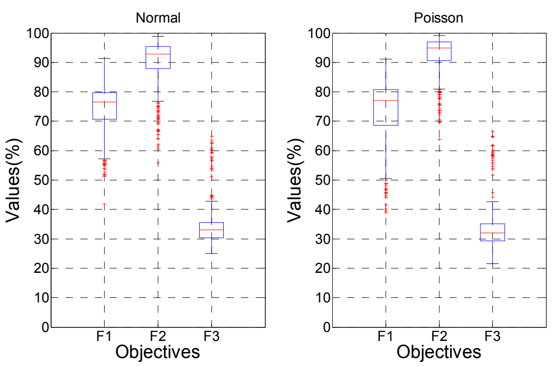

Figure 9. Box plots of non-dominated solutions for solving the 3D deployment problem with normal and Poisson detection levels, using 50 sensors.

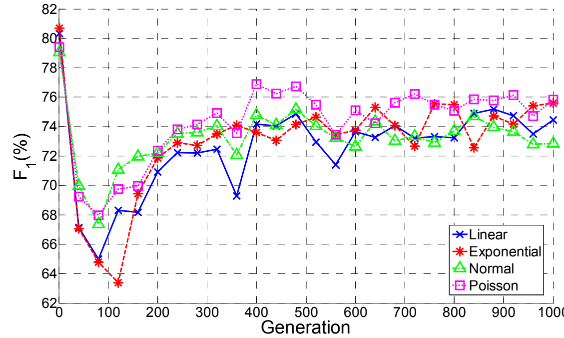

Figure 10. The mean objective value F_1 of non-dominated solutions in each generation, for four problems with different detection levels, using 50 sensors.

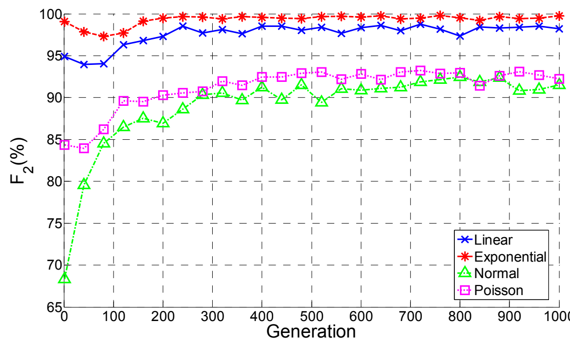

Figure 11. The mean objective value F_2 of non-dominated solutions in each generation, for four problems with different detection levels, using 50 sensors.

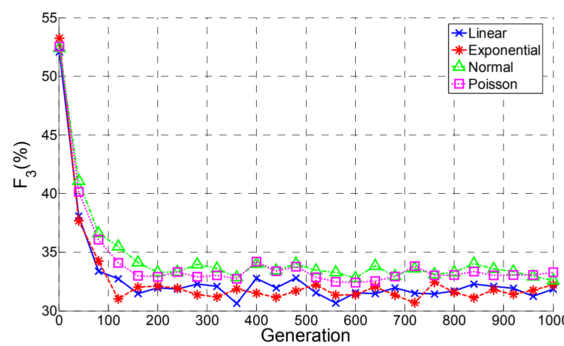

Figure 12. The mean objective value F_2 of non-dominated solutions in each generation, for four problems with different detection levels, using 20 sensors.

VI. CONCLUSION

In this paper, a multi-objective evolutionary approach is proposed to solve 3D differentiated WSN deployment problems. Experimental results demonstrated MOGA is capable of optimizing coverage, satisfaction of detection levels, and energy conservation. Moreover, MOGA can provide mission planers a set of non-dominated solutions for deployment of sensor nodes. The results also indicate that some problems with unusual detection levels requirements may require more sensor nodes for MOGA than those of problems with usual detection levels requirements. Our future work will develop specialized techniques for 3D WSN deployment problems with unusual detection levels.

ACKNOWLEDGMENT

This work was supported by the National Science Council of Taiwan, R.O.C. under Contract NSC-96-2221-E-216-037-MY2, and Chung-Hua University under Contract CHU-96-2221-E-216-037-MY2.

REFERENCES

[1] N. Aitsaadi, N. Achirt, K. Boussetta, and G. Pujolle, “A Tabu Search Approach for Differentiated Sensor Network Deployment,” in *Proc. of 5th IEEE Consumer Communications and Networking Conference, 2008. CCNC 2008*, pp. 163-167, 2008.

[2] N. Aitsaadi, N. Achir, K. Boussetta, and G. Pujolle, "Differentiated Underwater Sensor Network Deployment," in *Proc. of OCEANS 2007 – Europe*, pp.1-6, June 2007.

[3] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, “Grid coverage for surveillance and target location in distributed sensor networks,” *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1448-1453, 2002.

[4] M. Cardei, J. Wu, and M. Lu, “Improving network lifetime using sensors with adjustable sensing ranges,” *International Journal of Sensor Networks*, vol. 1, no. 1/2, pp. 41-49, 2006.

[5] A. Dhawan, C. T. Vu, A. Zelikovsky, Y. Li, and S. K. Prasad, “Maximum Lifetime of Sensor Networks with Adjustable Sensing Range,” in *Proc. of Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006*, pp. 285-289, 2006.

[6] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001.

[7] J. Jia, J. Chen, G.-R. Chang, and Y.-Y. Wen, “Efficient Cover Set Selection in Wireless Sensor Networks,” *Acta Automatica Sinica*, vol. 34, no. 9, pp. 1157-1162, 2008.

[8] H.-C. Jang and H.-C. Lee, “Efficient energy management to prolong wireless sensor network lifetime,” in *Proc. of 3rd IEEE/IFIP International Conference in Central Asia on Internet, ICI 2007.*, pp. 1-4, 2007.

[9] S. Oktug, A. Khalilov, and H. Tezcan, “3D Coverage Analysis under Heterogeneous Deployment Strategies in Wireless Sensor Networks,” in *Proc. of Fourth Advanced International Conference on Telecommunications, 2008. AICT '08*, pp. 199-204, 2008.

[10] S. C. Oh, C. H. Tan, F. W. Kong, Y. S. Tan, K. H. Ng, G. W. Ng, and K. Tai, “Multiobjective optimization of sensor network deployment by a genetic algorithm,” in *Proc. of IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, pp. 3917-3921, 2007.

[11] Y. Shang and H. Shi, “Coverage and energy tradeoff in density control on sensor networks,” in *Proc. of 11th International Conference on Parallel and Distributed Systems, 2005*, pp. 564-570, 2005.

[12] J. Wu and S. Yang, “Coverage issue in sensor networks with adjustable ranges,” in *Proc. of 2004 International Conference on Parallel Processing Workshops, 2004. ICPP 2004*, pp. 61-68, 2004.

[13] S. Zhang, Y. Liu, J. Pu, X. Zeng, and Z. Xiong, “An Enhanced Coverage Control Protocol for Wireless Sensor Networks,” in *Proc. of 42nd Hawaii International Conference on System Sciences, 2009. HICSS '09.*, pp. 1-7, 2009.

A Multi-Objective Evolutionary Approach for Combined Heat and Power Environmental/Economic Power Dispatch

Chien-Hung Chen and Jian-Hung Chen*
Department of Computer Science and Information Engineering
Chung-Hua University, Hsin-Chu 300, Taiwan
jh.chen@ieee.org*

Abstract— **Economic dispatch is to determine an efficient, low-cost and reliable operation of a power system by dispatching the available electricity generation resources to supply the demands. Traditionally, the primary objective of economic dispatch is to minimize the total cost of generation while satisfying the operational constraints of the available generation resources. However, due to environmental awareness and environmental policies, the design and operation of electric utilities are forced to restructure their power system systems to account for their emission impacts. In this paper, a combined heat and power system is considered. A combined heat and power environmental/economic dispatch (CHPEED) problem is formulated. Four objectives: fuel cost, emission, power overhead and heat overhead are considered in CHPEED problems. A multi-objective evolutionary approach is proposed to solve the CHPEED problems.**

Keywords-Cogeneration, Heat and power dispatch, Economic/environmental dispatch, multi-objective optimization, genetic algorithms

I. INTRODUCTION

Economic dispatch (ED) is to determine an efficient, low-cost and reliable operation of a power system by dispatching the available electricity generation resources to supply the demands in such a manner that the cost of operation is minimized while all operational constraints are satisfied. However, due to increasing concerns on environmental issues and the implementation of the Clean Air Act Amendments, environmental constraints have topped the list of utility management concerns. This issue that has attracted much attention is pollution minimization due to the pressing public demand for clean air. Therefore, operating power systems at absolute minimum fuel cost can no longer be the only criterion for dispatching electric power nowadays [1].

In the past decades, increasing demand for power and heat resulted in the existence of cogeneration units [2]. Cogeneration is also referred to as a combined heat and power (CHP) system. It produces electricity and useful heat simultaneously. Some industrial processes have large heat requirements, either as process steam or piped hot fluid, as well as large power demands [3]. Traditional, the primary objective of combined heat and power economic dispatch (CHPED) is similar to economic dispatch problems. The objective of CHPED is to find the optimal point of power and heat generation with minimum fuel cost such that both heat and power demands are met while the combined heat and power units are operated in a bounded heat versus power plane. The mutual dependencies of heat and power generation introduce a complication in the integration of cogeneration units into the power system economic dispatch [2].

The generation of power and heat from fossil fuel releases several contaminants, such as Sulfur Oxides, Nitrogen Oxides and Carbon Dioxide, into the atmosphere [4]. However, the increasing public awareness of the environmental protection has forced the utilities to modify their design or operational strategies to reduce pollution and environmental emissions of the thermal power plants [5]. Therefore, it becomes very complicated when dealing with increasingly complex dispatch problems for conventional techniques.

As a result, economic/environmental dispatch is a multi-objective problem with conflicting objectives because pollution minimization is conflicting with minimum cost of generation [1]. In this paper, a combined heat and power environmental/economic dispatch (CHPEED) problem, considering the fuel cost, emission, power overhead and heat overhead, is formulated. A multi-objective evolutionary approach is proposed in this paper to optimize these four objectives simultaneously.

II. RELATED WORK

A. Environmental/Economic Dispatch Problem

Environmental issue has become one of the most important factors in environmental/economic dispatch (EED) problem. Emissions are taken into consideration except fuel cost for it is more and more important to save environment from the pollutants caused by power plants. In [6], it treats the emission as a constraint with a permissible limit. This formulation, however, has a severe difficulty in getting the trade-off relations between cost and emission [5]. In [7-10], the emission is treated as another objective in addition to usual cost objective. However, the EED problem was converted to a single objective problem either by linear combination of both objectives or by considering one objective at a time for optimization. Unfortunately, this approach requires multiple runs as many times as the number of desired Pareto-optimal solutions and tends to find weakly non-dominated solutions [5]. In [11-13], both fuel cost and emission are taken into consideration simultaneously. The approach proposed in [11-13] handles both fuel cost and emission simultaneously as competing objectives. Stochastic search and fuzzy-based multi-objective optimization techniques have been proposed for the EED problem. However, the algorithms do not provide a systematic framework for directing the search towards Pareto-optimal front and the extension of these techniques to include more objectives is a very involved question. In addition, these techniques are computationally involved and time-consuming [5]. Genetic algorithm based multi-objective optimization techniques have been adopted in [14, 15] where a set of good non-dominated solutions can be obtained from each evolution generation. However, GA-based techniques suffer from premature convergence and the technique presented in [14] is computationally involved due to ranking process during the fitness assignment procedure. In [5], a new multi-objective particle swarm optimization (MOPSO) technique for environmental/economic dispatch (EED) problem is proposed. The proposed MOPSO technique evolves a multi-objective version of PSO by proposing redefinition of global best and local best individuals in multi-objective optimization domain.

When some industrial processes have large heat requirements, the heat load becomes as important as power load. As a result, the combined heat and power economic dispatch (CHPED) problem of a system has been raised to determine the unit heat and power production, so that the system production cost is minimized while the heat and power demands and other constraints are met. In [2], a self adaptive real-coded genetic algorithm (SARGA) is implemented to solve the problem. However, environmental emission is not considered in this paper.

Nevertheless, these EED and CHPED problems only considered a fixed number of power/cogeneration units or heat-alone units while optimizing fuel costs and emissions. None of them consider environmental/economic dispatch with a variable number of units.

B. *Multi-objective Evolutionary Optimization*

Assume the multi-objective functions are to be minimized. Mathematically, MOOPs can be represented as the following vector mathematical programming problems

$$Minimize \quad F(Y) = \{F_1(Y), F_2(Y), ..., F_i(Y)\}. \tag{1}$$

where Y denotes a solution and $f_i(Y)$ is generally a nonlinear objective function. Pareto dominance relationship and some related terminologies are introduced below. When the following inequalities hold between two solutions Y_1 and Y_2, Y_2 is a non-dominated solution and is said to dominate Y_1 ($Y_2 \succ Y_1$):

$$\forall i : F_i(Y_1) > F_i(Y_2) \wedge \exists j : F_j(Y_1) > F_j(Y_2). \tag{2}$$

When the following inequality hold between two solutions Y_1 and Y_2, Y_2 is said to weakly dominate Y_1 ($Y_2 \succeq Y_1$):

$$\forall i : F_i(Y_1) \geq F_i(Y_2). \tag{3}$$

A feasible solution Y^* is said to be a Pareto-optimal solution if and only if there does not exist a feasible solution Y where Y dominates Y^*, and the corresponding vector of Pareto-optimal solutions is called Pareto-optimal front.

By making use of Pareto dominance relationship, multi-objective evolutionary algorithms (MOEAs) are capable of performing the fitness assignment of multiple objectives without using relative preferences of multiple objectives. Thus, all the objective functions can be optimized simultaneously. As a result, MOEA seems to be an alternative approach to solving production planning and inspection planning problems on the assumption that no prior domain knowledge is available [13].

III. PROBLEM STATEMENT

The CHPEED problem is to minimize four competing objective functions, fuel cost, emission, power overhead and heat overhead, while satisfying several equality and inequality constraints. The CHPEED problem is formulated as follows.

A. *Problem objectives*

1) *Minimization of fuel cost*

The total US$/h fuel cost F_{cost} can be expressed as

$$F_{\cos t} = \sum_{i=1}^{N_p} C_i(P_i) + \sum_{j=1}^{N_c} C_j(O_j, H_j) + \sum_{k=1}^{N_h} C_k(T_k) \tag{4}$$

, where C_i, C_j and C_k are the unit production costs of the conventional power, cogeneration and heat-alone units, respectively; P_i and O_j are power generations of conventional power and cogeneration units; H_j and T_k are heat generation of cogeneration and heat-alone units.

2) *Minimization of emission*

$$E = \sum_{i=1}^{N_p} E_i(P_i) + \sum_{j=1}^{N_c} E_j(O_j, H_j) + \sum_{k=1}^{N_h} E_k(T_k) \tag{5}$$

, where E_i, E_j *and* E_k are the emission (kg/h) caused by the conventional power, cogeneration and heat-alone units, respectively.

$$E_i(P_i) = \alpha + \beta P_i + \gamma P_i^2 \tag{6}$$

$$E_j(O_j) = \mu O_j \tag{7}$$

$$E_k(T_k) = T_k(\mu_{NO_x} + \mu_{CO_2} + \mu_{CO}) \tag{8}$$

, where α, β *and* γ represent to the emission function coefficients of the conventional power unit.

3) *Minimization of power overhead and heat overhead*

$$O_p = \sum_{i=1}^{N_p} P_i + \sum_{j=1}^{N_c} O_j - P_d \quad (9)$$

$$O_h = \sum_{j=1}^{N_c} H_j + \sum_{k=1}^{N_h} T_k - H_d \quad (10)$$

, where H_d and P_d are heat and power demands; N_p, N_c and N_h denote the number of conventional power, cogeneration and heat-alone units, respectively.

B. Problem constraints

$$\sum_{i=1}^{N_p} P_i + \sum_{j=1}^{N_c} O_j \geq P_d \quad (11)$$

$$\sum_{j=1}^{N_c} H_j + \sum_{k=1}^{N_h} T_k \geq H_d \quad (12)$$

$$P_i^{\min} \leq P_i \leq P_i^{\min}, \quad i = 1, \ldots, N_p \quad (13)$$

$$O_j^{\min}(H_j) \leq O_j \leq O_j^{\max}(H_j), \quad j = 1, \ldots, N_c \quad (14)$$

$$H_j^{\min}(O_j) \leq H_j \leq H_j^{\max}(O_j), \quad j = 1, \ldots, N_c \quad (15)$$

$$T_k^{\min} \leq T_k \leq T_k^{\max}, \quad k = 1, \ldots, N_h \quad (16)$$

with

$$C_i(P_i) = a_p + b_p P_i + c_p P_i^2 \quad (17)$$

$$C_j(O_j, H_j) = a_c + b_c O_j + c_c O_j^2 + d_c H_j + e_c H_j^2 + f_c O_j H_j \quad (18)$$

$$C_k(T_k) = a_h + b_h T_k + c_h T_k^2 \quad (19)$$

, where P_i^{min} and P_i^{max} are the minimum and maximum power generation limits of the conventional units; O_j^{min} and O_j^{max} are the minimum and maximum power generation limits of the cogeneration units; H_j^{min} and H_j^{max} are the minimum and maximum heat generation limits of the cogeneration units; T_k^{min} and T_k^{max} are the minimum and maximum heat generation limits of the heat-alone units; a_p, b_p and c_p are fuel cost coefficients of the conventional power unit ; a_c , b_c , c_c , d_c , e_c and f_c are fuel cost coefficients of the cogeneration unit; a_h, b_h and c_h are fuel cost coefficients of the heat-alone unit. The value of fuel cost coefficients are given in Table I.

TABLE I. GENERATOR FUEL COST COEFFICIENTS.

coefficients	**unit**		
	Conventional power	***Cogeneration***	***Heat-alone***
a	451.32513	2650	0
b	46.15916	14.5	$23.4T_1$
c	0.10587	0.0345	0
d		4.2	
e		0.03	
f		0.031	

IV. MULTI-OBJECTIVE GENETIC ALGORITHM

A. Chromosome Representation

A chromosome has gene information for solving the problem in MOGA. Each chromosome has dynamic gene size, which is determined by the max number of all units in combined heat and power (CHP) systems. The first gene is numbers of conventional power unit and the second one stands for numbers of cogeneration, and the third one represents numbers of heat-alone unit. The remains of the genes are the dispatch value of all units.

B. Fitness Assignment

We use a generalized Pareto-based scale-independent fitness function (GPSIFF) considering the quantitative fitness values in Pareto space for both dominated and non-dominated individuals. GPSIFF makes the best use of Pareto dominance relationship to evaluate individuals using a single measure of performance. The used GPSIFF is briefly described below. Let the fitness value of an individual *Y* be a tournament-like score obtained from all participant individuals by the following function:

$$F(Y) = p - q + c \quad (20)$$

, where p is the number of individuals which can be dominated by the individual Y, and q is the number of individuals which can dominate the individual Y in the objective space. Generally, a constant c can be optionally added in the fitness function to make fitness values positive. c is usually set to the number of all participant individuals.

C. Genetic Operators

The genetic operators used in the proposed approach are widely used in literature. The selection operator uses a binary tournament selection without replacement, which works as follows. Choose two individuals randomly from the population and copy the better individual into the intermediate population.

Crossover is a recombination process in which genes from two selected parents are recombined to generate offspring chromosomes. The order crossover (OX) in GA literature is used in our approach.

A simple mutation operator is used to alter genes. For each gene, randomly generate a real value from their given range. If the value is smaller than the mutation probability p_m, replace its index with a randomly generated integer among its possible values.

D. Procedure of MOGA

The procedure of MOGA is written as follows:

Input: population size N_{pop}, recombination probability p_c, mutation probability p_m, the number of maximum generations G_{max}.

Output: The optimum solutions ever found in P.

Step 1: Initialization Randomly generate an initial population P of N_{pop} individuals.

Step 2: Evaluation For each individual in the population, compute all objective function values F_1, F_2, and F_3.

Step 3: Fitness assignment Assign each individual a fitness value by using GPSIFF.

Step 4: Selection Select N_{pop} individuals from the population to form a new population using the binary tournament selection.

Step 5: Recombination Perform the order crossover operation with a recombination probability p_c.

Step 6: Mutation Apply the mutation operator to each gene in the individuals with a mutation probability p_m.

Step 7: Termination test If a stopping condition is satisfied, stop the algorithm. Otherwise, go to Step 2.

V. RESULTS AND DISCUSSIONS

A. Simulation Environment and Parameter Settings

This power system considers a type of conventional power unit, cogeneration unit and heat-alone unit, respectively. The power generation limits of the conventional power unit are 0 and 150 MW and heat generation limits of heat-alone units are 0 and 2695.2 MW_{th}. The feasible operating regions of the cogeneration unit are given in figure 1. The value of emission coefficients α, β *and* γ are given as 13.85932, 0.32767 and 0.00419, respectively. The emission factors of heat-alone units are obtained from the average heat generation from residential boilers in urban areas, with an equivalent fuel mix as input [16]. The emission factors μ_{NOx}, μ_{CO2} *and* μ_{CO} are given as 0.2 kg/MW, 0.27 kg/MW and 0.04 kg/MW, respectively.

The feasible operating regions of the cogeneration unit from Figure 1 can be expressed as inequality constraints as follows:

$$1.781914894H - O - 105.74468090 \leq 0 \quad (21)$$

$$0.177777778H + O - 247.0 \leq 0 \quad (22)$$

$$-0.169847328H - O + 98.8 \leq 0 \quad (23)$$

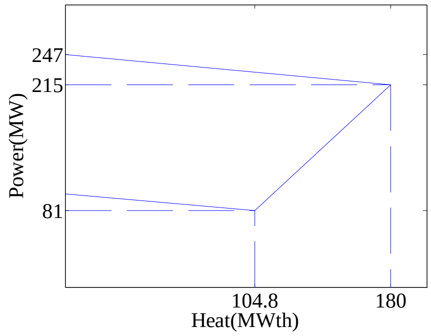

Figure 1. Feasible operating regions of cogeneration unit.

Based on the given environment and constraints, three benchmark problems “demand (200, 115)”, “demand (700, 615)” and “demand (2000, 1115)” are designed to validate our approach. The notation “demand (P, H)” represents that the power demand is P and the heat demand is H.

The parameter settings of MOGA are listed as follows: population size N_{pop}=50, recombination probability p_c=0.9, mutation probability p_m=0.01, the number of maximum generations G_{max}=100. Thirty independent runs are conducted for each problem.

Figures 2-4 shows the distributions of non-dominated solutions in four objectives by means of boxplot. The results indicate that the proposed approach is capable of obtaining a set of wide-spread and non-dominated solutions.

Figures 5-8 depict a typical run of MOGA in solving “demand (2000, 1115)”. The maximum, mean and minimum objective values of individuals during a typical run are shown in the figures. The results indicate that the proposed approach converge steadily and rapidly.

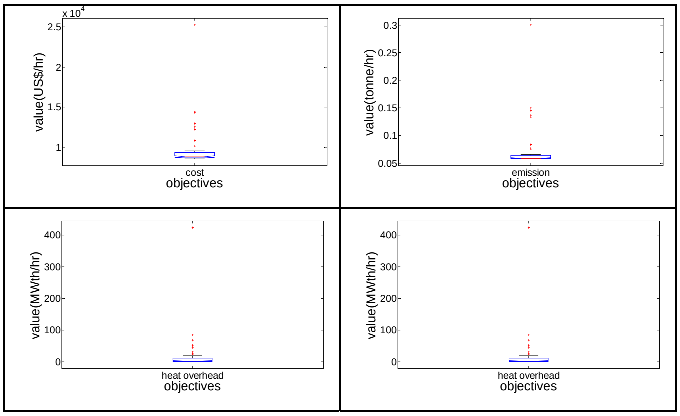

Figure 2. Boxplot of non-dominated solutions in solving “demand (200,115)” problem.

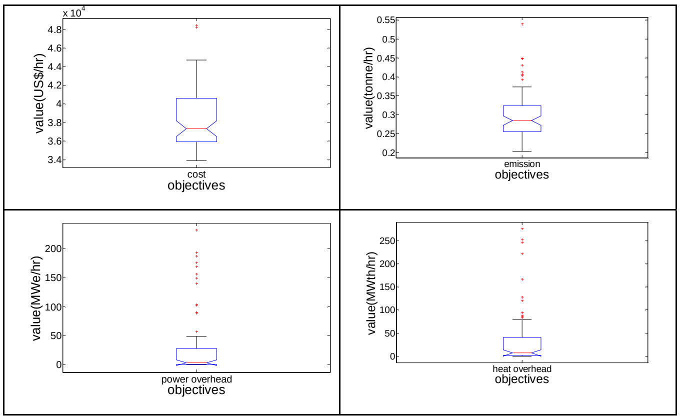

Figure 3. Boxplot of non-dominated solutions in solving "demand (700,615)" problem.

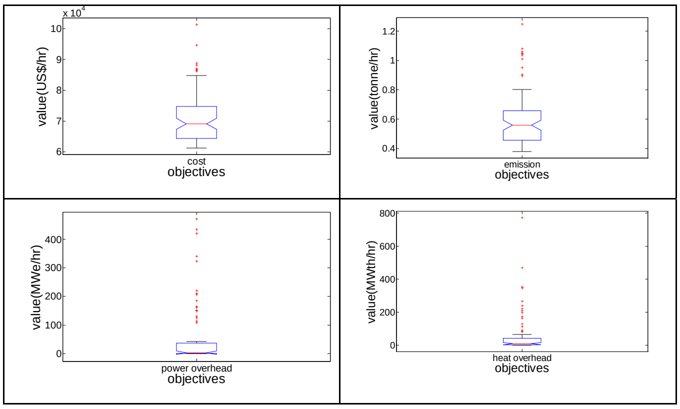

Figure 4. Boxplot of non-dominated solutions in solving "demand (2000,1115)" problem.

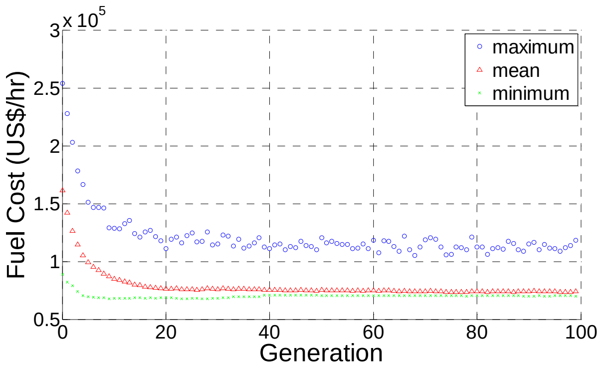

Figure 5. The maximum, mean, and minimum fuel cost of a typical run in solving "demand (2000,1115)" problem.

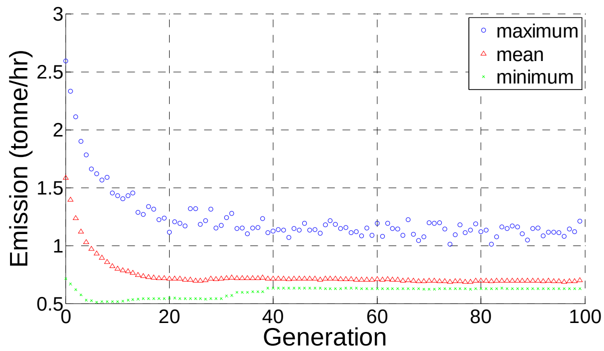

Figure 6. The maximum, mean, and minimum emission of a typical run in solving "demand (2000,1115)" problem.

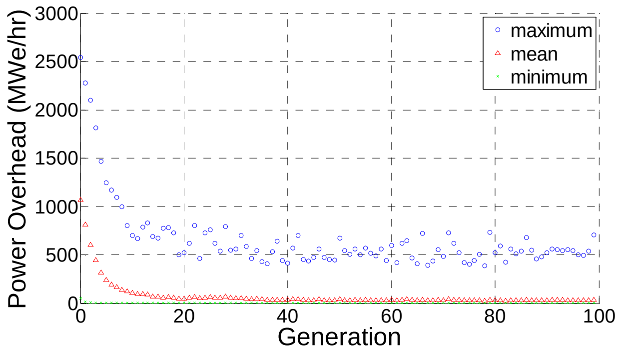

Figure 7. The maximum, mean, and minimum power overhead of a typical run in solving "demand (2000,1115)" problem.

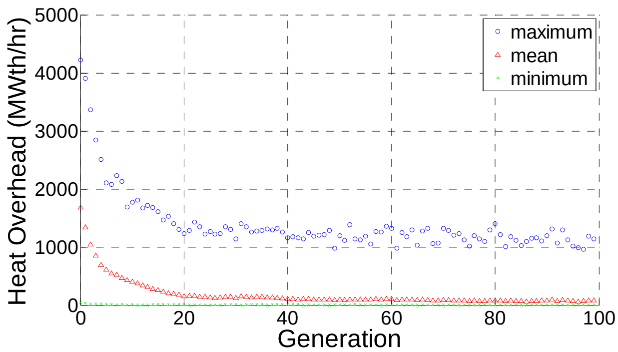

Figure 8. The maximum, mean, and minimum heat overhead of a typical run in solving "demand (2000,1115)" problem.

VI. CONCLUSION

In this paper, a multi-objective evolutionary approach is proposed to solve the combined heat and power environmental/economic dispatch problem. The problem is formulated as multi-objective optimization problem with competing economic and environmental objectives. Experimental results demonstrated the proposed method is capable of optimizing fuel cost, emission, power overhead and heat overhead simultaneously. Moreover, the proposed approach can provide decision makers a set of non-dominated solutions to choose a suitable dispatch plan.

ACKNOWLEDGMENT

This work was supported by the National Science Council of Taiwan, R.O.C. under Contract NSC-96-2221-E-216-037-MY2, and Chung-Hua University under Contract CHU-96-2221-E-216-037-MY2.

REFERENCES

[1] L. Xuebin, "Study of multi-objective optimization and multi-attribute decision-making for economic and environmental power dispatch," *Electric Power Systems Research,* vol. 79, pp. 789-795, 2009.

[2] P. Subbaraj, R. Rengaraj, and S. Salivahanan, "Enhancement of combined heat and power economic dispatch using self adaptive real-coded genetic algorithm," *Applied Energy,* vol. 86, pp. 915-921, 2009.

[3] K. P. Wong and C. Algie, "Evolutionary programming approach for combined heat and power dispatch," *Electric Power Systems Research,* vol. 61, pp. 227-232, 2002.

[4] R. Ramanathan, "Emission constrained economic dispatch," *Power Systems, IEEE Transactions on,* vol. 9, pp. 1994-2000, 1994.

[5] M. A. Abido, "Multiobjective particle swarm optimization for environmental/economic dispatch problem," *Electric Power Systems Research,* vol. 79, pp. 1105-1113, 2009.

[6] G. P. Granelli, M. Montagna, G. L. Pasini, and P. Marannino, "Emission constrained dynamic dispatch," *Electric Power Systems Research,* vol. 24, pp. 55-64, 1992.

[7] A. Farag, S. Al-Baiyat, and T. C. Cheng, "Economic load dispatch multiobjective optimization procedures using linear programming techniques," *Power Systems, IEEE Transactions on,* vol. 10, pp. 731-738, 1995.

[8] J. S. Dhillon, S. C. Parti, and D. P. Kothari, "Stochastic economic emission load dispatch," *Electric Power Systems Research,* vol. 26, pp. 179-186, 1993.

[9] C. S. Chang, K. P. Wong, and B. Fan, "Security-constrained multiobjective generation dispatch using bicriterion global optimisation," *Generation, Transmission and Distribution, IEE Proceedings-,* vol. 142, pp. 406-414, 1995.

[10] R. Yokoyama, S. H. Bae, T. Morita, and H. Sasaki, "Multiobjective optimal generation dispatch based on probability security criteria," *Power Systems, IEEE Transactions on,* vol. 3, pp. 317-324, 1988.

[11] D. Srinivasan, C. S. Chang, and A. C. Liew, "Multiobjective generation scheduling using fuzzy optimal search technique," *Generation, Transmission and Distribution, IEE Proceedings,* vol. 141, pp. 233-242, 1994.

[12] H. Chao-Ming, Y. Hong-Tzer, and H. Ching-Lien, "Bi-objective power dispatch using fuzzy satisfaction-maximizing decision approach," *Power Systems, IEEE Transactions on,* vol. 12, pp. 1715-1721, 1997.

[13] D. B. Das and C. Patvardhan, "New multi-objective stochastic search technique for economic load dispatch," *Generation, Transmission and Distribution, IEE Proceedings-,* vol. 145, pp. 747-752, 1998.

[14] M. A. Abido, "A novel multiobjective evolutionary algorithm for environmental/economic power dispatch," *Electric Power Systems Research,* vol. 65, pp. 71-81, 2003.

[15] M. A. Abido, "Environmental/economic power dispatch using multiobjective evolutionary algorithms," *Power Systems, IEEE Transactions on,* vol. 18, pp. 1529-1537, 2003.

[16] G. Chicco, P. Mancarella, and R. Napoli, "Emission Assessment of Distributed Generation in Urban Areas," in *Power Tech, 2007 IEEE Lausanne*, 2007, pp. 532-537.

A Force-Driven Evolutionary Approach for Multi-objective 3D Differentiated Sensor Network Deployment

Liang-Che Wei, Chih-Wei Kang and Jian-Hung Chen*
Department of Computer Science and Information Engineering
Chung-Hua University, Hsin-Chu 300, Taiwan
*jh.chen@ieee.org**

ABSTRACT

This paper describes a novel force-driven evolutionary approach for solving multi-objective 3D deployment problems in differentiated wireless sensor networks (WSNs). WSN is a wireless network consisting of spatially distributed autonomous sensors to monitor physical or environmental conditions. Deciding the location of sensor to be deployed on a terrain with the consideration of different criteria is an important issue for the design of wireless sensor network. A multi-objective genetic algorithm with a force-driven method is proposed to solve 3D differentiated WSN deployment problems with the objectives of the coverage of sensors, satisfaction of detection levels, and energy conservation. The preliminary experimental results demonstrated that the proposed approach is capable of obtaining a set of non-dominated solutions for multi-objective 3D differentiated WSN deployment problems.

1. INTRODUCTION

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous sensors to monitor physical or environmental conditions. Sensor nodes of a WSN are deployed over a region to sense events on geographical areas and transmit collected data to a sink node for further operations. Depending on the requirements, sensors could be deployed in diverse scenarios [4,9]. Therefore, deciding the location of sensor to be deployed on a terrain is an important issue. Several different objectives should be considered and fulfilled in the design phase of WSNs, such as the coverage and accuracy, reaction time and survivability of the sensor network. However, these objectives may be in conflict with one another and of different importance to mission planners [10].

978-1-4244-5113-5/09/$25.00 (c)2009 IEEE

Coverage is one of the fundamental issue in the deployment of WSNs. WSNs have to maintain sufficient coverage quality in order to capture the timely changing targets [13]. For enhanced coverage, a large number of sensors are typically deployed in the sensor field and, if the coverage areas of multiple sensors overlap, they may all report a target in their respective zones [3].

Differentiated sensor network deployment, which considers the satisfaction of detection levels in different geographical characteristics, is also an important issue [1]. In some specially designated WSN applications, such as underwater sensor deployment, mudflows and landslide monitoring, depending on the event's location, the supervised area may require different detection levels. Therefore, the sensing requirements of these applications are not uniformly distributed within the area. As a result, the deployment strategy of WSN should take into consideration the geographical characteristics of the monitored events.

Energy conservation for the lifetime of sensors is another rising issue [5]. Due to the limited energy resource in each sensor node, utilizing sensors in an efficient manner so as to increase the lifetime of the network is an important task in the design phase of WSNs. There are two different approaches: scheduling and adjusting methods, to the problem of conserving energy in sensor networks. We focus on adjusting the sensing range of each sensor in order to reduce the overlaps among sensing ranges while keep the detection ability above a predefined detection level.

In this paper, a 3D differentiated WSN deployment problem is formulated into a multi-objective optimization problem. Three objectives are to be optimized: maximizing coverage of sensors, satisfying the required probability of detection level, and minimizing the detection power by adjustable sensing range. A multi-objective genetic algorithm (MOGA) framework with a novel force-driven method is proposed to solve these problems.

2. RELATED WORK

2.1. WSN Deployment Problem

Coverage issue is one of the most important tasks in WSN. The ultimate goal is to have each location in the physical space of interest within the sensing range of at least one sensor. However, due to the number of sensors is limited, complete coverage cannot be guaranteed. Therefore, many approaches are proposed to deal with the 2D coverage problem [7, 10]. Recently, Oktug et al. [9] proposed an approach to solve coverage problem by simulating sensor deployment strategies on a 3D terrain model and to find answers to questions that how many sensors are needed to cover a specified 3D terrain at a specified coverage percentage.

Different applications require different degrees of sensing coverage. While some applications may require a complete coverage in a region, others may only need a high percentage of coverage. Such WSN is called differentiated WSN [1]. Take underwater sensor deployment [2] as an example, sensor field of underwater is characterized by the geographical irregularity of the sensed events because some area may be inaccessible or the event area may not be uniformly distributed. To efficiently monitor such area with differentiated detection levels, fulfillment of detection levels in different area is the major concerns instead of maximizing the coverage of sensors [11]. Aitsaadi et al. [1] proposed a probabilistic event detection model. In this model, each grid point has a required minimum probability detection threshold. A tabu search method is proposed to solve this differentiated WSN deployment problem.

In recent years, utilizing limited energy efficiently in a wireless sensor network has become an important issue. Several techniques, such as scheduling models and sleep models [4, 8, 12], have been proposed to extend the lifetime of WSNs.

2.2. Multi-objective Evolutionary Optimization

Assume the multi-objective functions are to be minimized. Mathematically, multi-objective optimization problems (MOOPs) can be represented as the following vector mathematical programming problems

$$Minimize \quad F(Y) = \{F_1(Y), F_2(Y), ..., F_i(Y)\}, \tag{1}$$

where Y denotes a solution and $F_i(Y)$ is generally a nonlinear objective function. Pareto dominance relationship and some related terminologies are introduced below. When the following inequalities hold between two solutions Y_1 and Y_2, Y_2 is a non-dominated solution and is said to dominate Y_1 ($Y_2 \succ Y_1$):

$$\forall i : F_i(Y_1) \geq F_i(Y_2) \wedge \exists j : F_j(Y_1) > F_j(Y_2). \tag{2}$$

When the following inequality hold between two solutions Y_1 and Y_2, Y_2 is said to weakly dominate Y_1 ($Y_2 \succeq Y_1$):

$$\forall i : F_i(Y_1) \geq F_i(Y_2). \tag{3}$$

A feasible solution Y^* is said to be a Pareto-optimal solution if and only if there does not exist a feasible solution Y where Y dominates Y^*.

By making use of Pareto dominance relationship, multi-objective evolutionary algorithms (MOEAs) [6] are capable of performing the fitness assignment of multiple objectives without using relative preferences of multiple objectives.

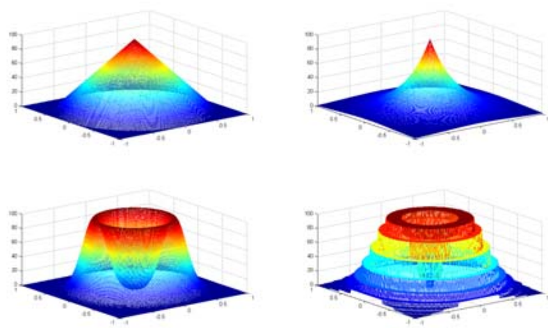

Figure 1. Terrain with different required detection levels: decreasing linear, normal, Poisson, and exponential distributions.

3. PROBLEM STATEMENT

3.1. Notations

In order to formulate problems, the following notations are introduced:

- i : sensor index, $i = 1,2,3,\ldots,N$.
- j : grid point index, $j = 1,2,3,\ldots,M$.
- k : sensing range index, $k = 1,2,3,\ldots,K$.

3.2. Environment

We assume that N sensors $s_1, s_2, \ldots, s_N$ are deployed to cover the sensor field. Let the sensor field T consist of n_x, n_y, and n_z grid points $p_1, p_2, \ldots, p_M$ in the x, y, and z dimensions, respectively [3]. Each sensor has an initial sensor energy E and has the capability to adjust its sensor range. Sensing range options are $r_1, r_2, \ldots, r_K$, corresponding to energy consumptions of $e_1, e_2, \ldots, e_K$ and detection error ranges $f_1, f_2, \ldots, f_K$ ($f_k < r_k$) [4]. We assume that each grid point p_j in sensor field is associated a required minimum probability detection level, denoted $t(p_j)$.

3.3. Mathematical Formation of 3D Deployment Problem

3.3.1. Maximization of Coverage.

In many WSN applications, the main task is the surveillance of certain geographical areas [9]. Target location can be simplified considerably if the sensors are placed in such a way that every grid point in the sensor field is covered by sensors [3]. Assume that sensor s_i is deployed at grid point. For any grid point p_j, the Euclidean distance between sensor s_i and grid point p_j is denoted as

$$d(s_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{4}$$

, where x_i, x_j, y_i, y_j, z_i and z_j are coordinate location values. The following equation shows a binary coverage model expressing the coverage $c_b(s_i, p_j)$ of a grid point p_j by sensor s_i.

$$c_b(s_i, p_j) = \begin{cases} 1, & if\ d(s_i, p_j) < r_k(s_i) \\ 0, & otherwise \end{cases} \tag{5}$$

, where $r_k(s_i)$ is the sensing range of the sensor s_i.

The coverage rate optimization problem F_1 can be defined by

$$Max. \quad F_1 = \frac{\sum_{j=1}^{M} c_b(p_j)}{M} \tag{6}$$

, where $c_b(p_j)$ is the coverage of all sensors at grid point p_j by the Equation (5). This objective is to be maximized.

3.3.2. Maximization of Differentiated Detection Levels.

Considering differentiated detection levels, assumed that each grid point p_j in sensor field T is associated a required minimum detection level $t(p_j)$. A terrain may have different required detection levels, as illustrated in Figure 1. A good deployment for differentiated WSN should satisfy the following condition: for each p_j in T, the measured detection probability of p_j should be greater than or equal to $t(p_j)$ [1].

A probabilistic detection model for sensor deployment [1] is adopted into our model. Assume that event detection probability of a sensor diminishes as its distance to the sensed point increases. A probabilistic detection model of sensors is expressed as

$$c_p(s_i, p_j) = \begin{cases} 0, & \text{if } r_k(s_i) + f_k(s_i) \le d(s_i, p_j) \\ e^{-\lambda \alpha^{\beta}}, & \text{if } r_k(s_i) - f_k(s_i) < d(s_i, p_j) < \\ & \quad r_k(s_i) + f_k(s_i) \\ 1, & \text{if } r_k(s_i) - f_k(s_i) \ge d(s_i, p_j) \end{cases} \tag{7}$$

, where $\alpha = d(s_i, p_j) - (r_k(s_i) - f_k(s_i))$, λ and β are parameters that measure the detection probabilities when an object is within a certain distance from the sensor, and $f_k(s_i)$ is the error ranges of the sensor s_i. Each sensor s_i has a detection probability $c_p(s_i, p_j)$ at grid point p_j. A grid point p_j might be covered by more than one detection range of different sensors [8]. When a detection area is overlapped by multiple sensors, the closer are the sensors to each other, the higher is the detection probability of the grid points [7]. The conjunctive detection probability of all sensors at grid point p_j is given by

$$c_p(p_j) = 1 - \prod_{i=1}^{N} (1 - c_p(s_i, p_j)). \tag{8}$$

The optimization of the satisfaction required probability of detection level F_2 is expressed by:

$$Max. \quad F_2 = \frac{\sum_{j=1}^{M} DP(p_j)}{\sum_{j=1}^{M} t(p_j)} \tag{9}$$

,where $DP(p_j) = \begin{cases} t(p_j) & if\ c_p(p_j) - t(p_j) \ge 0 \\ 0 & otherwise \end{cases}$.

This objective is to be maximized.

3.3.3. Minimization of Energy Consumption

In terms of energy consumption, we only consider the energy used in sensing, but not including the power consumed by radio communication and computation. The sensing ranges of a sensor determine the energy consumed by the sensor [4]. We adopted an energy model in our evaluation. The power consumption is proportional to the square of the sensing range r_k [11]. The energy consumption model is expressed as follows:

$$e_k(s_i) = \mu \times r_k(s_i)^2, \tag{10}$$

where μ is an energy consumption parameter. The optimization of the detection power minimization with adjustable sensing range F_3 can be formulated as

$$Min. \quad F_3 = \frac{\sum_{i=1}^{N} e_k(s_i)}{\sum_{i=1}^{N} e_{\max}(s_i)} \tag{11}$$

, where $e_{max}(s_i)$ is the maximum detection range of each sensor. This objective is to be minimized.

4. FORCE-DRIVEN MULTI-OBJECTIVE GENETIC ALGORITHM (FD-MOGA)

4.1. Chromosome Representation

A chromosome has gene information for solving the problem in FD-MOGA. Each chromosome has fixed gene size, which is determined by the number of sensors in the WSN. Each gene has a *x, y,* and *z* coordinate location and a sensing range. The ranges of each gene of coordinate location are [0, n_x], [0, n_y], and [0, n_z] in the *x, y,* and *z* dimensions. Hence these sensors will have coordinate values to denote their location. Each gene of sensing range is one of $r_1, r_2, \ldots, r_K$, which represent the detection ability of the sensor.

4.2. Fitness Assignment

We use a generalized Pareto-based scale-independent fitness function (GPSIFF) considering the quantitative fitness values in Pareto space for both dominated and non-dominated individuals. Let the fitness value of an individual *Y* be a tournament-like score obtained from all participant individuals by the following function:

$$F(Y) = p - q + c \tag{12}$$

, where *p* is the number of individuals which can be dominated by the individual *Y*, and *q* is the number of individuals which can dominate the individual *Y* in the objective space. *c* is set to the number of all participant individuals.

4.3. Genetic Operators

The genetic operators used in the proposed approach are widely used in literature. The selection operator uses a binary tournament selection without replacement. The uniform crossover is used in FD-MOGA. A simple mutation operator is used to alter genes. For each gene, randomly generate a real value from the range [0, 1]. If the value is smaller than the mutation probability p_m, replace its index with a randomly generated integer among its possible values.

4.4. Repulsion and Attraction Force Mutation

To prevent sensors from overly centering in some positions in individuals, a force-driven method is introduced. The proposed force-driven method consists of two forces: repulsion force and attraction force. While the density of sensors within a certain space is high, a repulsion force mutation is to increase the degree of spread between sensors. On the contrary, while the density of sensors is low, an attraction force mutation is used to centralize sensors within a certain space. The procedure of repulsion and attraction force mutation is written as follows:

Step 1: Space Division Divide the sensor field *T* into bn_x, bn_y, and bn_z large grid space $bp_1, bp_2, \ldots, bp_L$, where $n_x > bn_x$, $n_y > bn_y$, and $n_z > bn_z$.

Step 2: Position Compute the position of sensors within each large grid space bp_l, $l = 1,2,\ldots, L$. Partition the sensors within the large grid space bp_l into a set S_l.

Step 3: Statistics Calculate the number of sensors, b_l, in each set S_l .

Step 4: Repulsion Mutation If the number b_l of sensors in a large grid space bp_l is bigger than one, repulse the positions of sensors in S_l from their centroid with one grid point in every dimension, and increase one level of sensing range in these sensor.

Step 5: Attraction Mutation If the number b_l of sensors in large grid space bp_l is equal to one, let the sensors adjacent to the large grid space bp_l be attracted and move to the position of the sensor in S_l with one grid point for every dimension, and decrease one level of sensing range in these sensors.

4.5. Procedure of FD-MOGA

An elitism strategy is adopted. An elite set *E* with capacity E_{max} will maintain all the best non-dominated solutions generated so far. The procedure of FD-MOGA is written as follows:

Input: population size N_{pop}, recombination probability p_c, mutation probability p_m, the number of maximum generations G_{max}.

Output: The optimum solutions ever found in *P*.

Step 1: Initialization Randomly generate an initial population *P* of N_{pop} individuals, and create an empty elite sets *E*.

Step 2: Evaluation For each individual in the population, compute all objective function values F_1, F_2, and F_3.

Step 3: Fitness assignment Assign each individual a fitness value by using GPSIFF.

Step 4: Update elitist Add the non-dominated individuals in *E*. Considering all individuals in *E*, remove the dominated ones in *E*. If the number of non-dominated individuals in *E* is larger than E_{max}, randomly discard excess individuals.

Step 5: Selection Select N_{pop} - N_{ps} individuals from the population to form a new population using the binary tournament selection and random select N_{ps} individuals from E to form a new population, where $N_{ps} = N_{pop} \times p_s$ and p_s is a selection proportion. If N_{ps} is greater than the number N_E of individuals in E, let $N_{ps} = N_E$..

Step 6: Recombination Perform the uniform crossover operation with a recombination probability p_c.

Step 7: Mutation Apply the simply mutation operator to each gene in the individuals with a mutation probability p_m.

Step 8: Repulsion and Attraction Mutation Execute the repulsion and attraction mutation to each individual with two probabilities p_r and p_a.

Step 9: Termination test If a stopping condition is satisfied, stop the algorithm. Otherwise, go to Step 2.

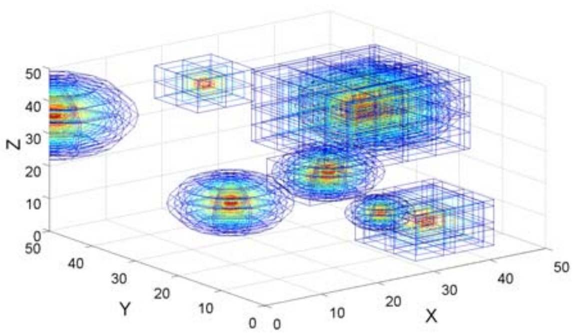

Figure 2. A terrain with decreasing linear detection levels.

5. RESULT AND DISCUSSION

5.1. Simulation Environment and Parameters

A 3D WSN deployment benchmark generator for WSN environment is designed to generate different scale of sensor fields with different models of detection probability levels. A sensor field with 50×50×50 grid points is generated. The same terrain with four different required minimum detection probability levels: decreasing linear, normal, Poisson, and exponential distributions, are illustrated as four different benchmarks. Figure 2 illustrates a terrain with linear decreasing levels. For the sensors of WSN, we assume each sensor has five adjustable sensing ranges 6, 8, 10, 12, 14, and the detection error ranges are half of the sensing range of each sensor. The power consumption parameter μ is 1. The probabilistic detection model parameter β is 0.5 and the detection radio wave parameter λ is 0.5.

The parameter settings of the proposed algorithm are listed as follows: population size N_{pop}=200, maximum number elite set of individuals E_{max}=10000, selection elite set proportion p_s=0.2, division of large grid space 5×5×5, recombination probability p_c=0.9, mutation probability p_m=0.01, repulsion probability p_r=0.1, attraction probability p_a=0.1, the number of maximum generations G_{max}=500 and 1000. Thirty independent runs are conducted for each problem. The number of sensor nodes to be deployed is limited to 20.

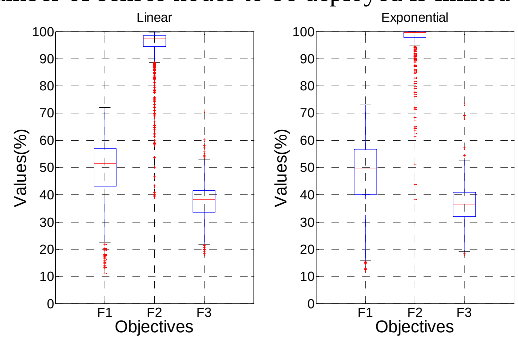

Figure 3. Box plots of non-dominated solutions for solving the 3D deployment problem with linear and exponential detection levels, using 20 sensors.

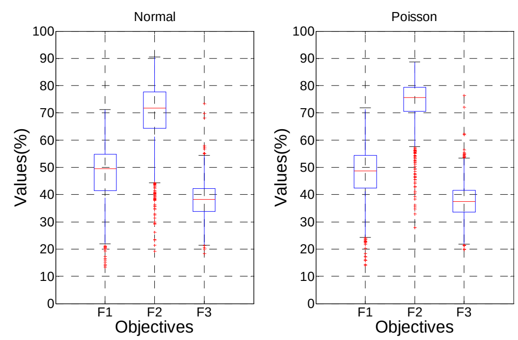

Figure 4. Box plots of non-dominated solutions for solving the 3D deployment problem with normal and Poisson detection levels, using 20 sensors.

Figures 3-4 depict the box plots of obtained non-dominated solutions. The results indicate that different detection levels pose different difficulties for FD-MOGA. The problems with normal and Poisson detection levels are more difficult to find a good deployment plan than problems with decreasing linear and exponential detection levels using the same number of sensors. The number of sensors required for a terrain with normal and Poisson detection levels should be bigger than the same terrain with decreasing linear and exponential detection levels.

A naïve MOGA without elitism and repulsion and attraction mutation is also implemented. The coverage metric *C(A,B)* of two solution sets A and B [6] used to compare the performance of two corresponding algorithms, FD-MOGA and MOGA, considering all the objectives.

$$C(A,B)=\frac{\{a\in A,b\in B,a\succeq b\}}{|B|}. \tag{13}$$

The value $C(A, B)$=1 means that all individuals in B are weakly dominated by A. Figure 5 depict box plots of coverage metric of FD-MOGA and MOGA in solving the 3D deployment problems with four detection levels, using 20 sensors. The result demonstrates the effectiveness of the elitism and force-driven mutation used in FD-MOGA.

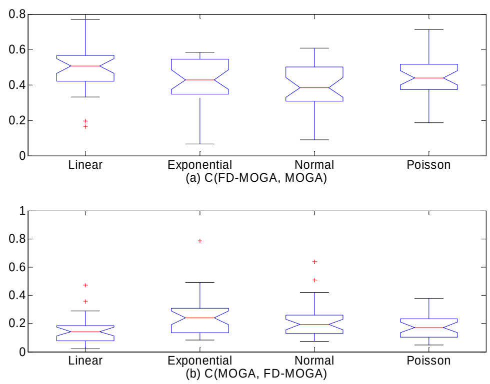

Figure 5. Box plots of coverage metric of FD-MOGA and MOGA for solving the 3D deployment problems with four detection levels, using 20 sensors.

6. CONCLUSION

In this paper, a force-driven multi-objective evolutionary approach is proposed to solve 3D differentiated WSN deployment problems. Experimental results demonstrated FD-MOGA is capable of optimizing coverage, satisfaction of detection levels, and energy conservation. Moreover, FD-MOGA can provide mission planers a set of non-dominated solutions for deployment of sensor nodes. The results also indicate that some problems with unusual detection levels requirements may require more sensor nodes for FD-MOGA than those of problems with usual detection levels requirements. Our future work will develop specialized techniques for 3D WSN deployment problems with unusual detection levels.

7. ACKNOWLEDGMENTS

This work was supported by the National Science Council of Taiwan, R.O.C. under Contract NSC-96-2221-E-216-037-MY2, and Chung-Hua University under Contract CHU-96-2221-E-216-037-MY2.

8. REFERENCES

[1] N. Aitsaadi, N. Achirt, K. Boussetta, and G. Pujolle, “A Tabu Search Approach for Differentiated Sensor Network Deployment,” in *Proc. of 5th IEEE Consumer Communications and Networking Conference, 2008. CCNC 2008*, pp. 163-167, 2008.

[2] N. Aitsaadi, N. Achir, K. Boussetta, and G. Pujolle, "Differentiated Underwater Sensor Network Deployment," in *Proc. of OCEANS 2007 – Europe*, pp.1-6, June 2007.

[3] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, “Grid coverage for surveillance and target location in distributed sensor networks,” *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1448-1453, 2002.

[4] M. Cardei, J. Wu, and M. Lu, “Improving network lifetime using sensors with adjustable sensing ranges,” *International Journal of Sensor Networks*, vol. 1, no. 1/2, pp. 41-49, 2006.

[5] A. Dhawan, C. T. Vu, A. Zelikovsky, Y. Li, and S. K. Prasad, “Maximum Lifetime of Sensor Networks with Adjustable Sensing Range,” in *Proc. of Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006*, pp. 285-289, 2006.

[6] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001.

[7] J. Jia, J. Chen, G.-R. Chang, and Y.-Y. Wen, “Efficient Cover Set Selection in Wireless Sensor Networks,” *Acta Automatica Sinica*, vol. 34, no. 9, pp. 1157-1162, 2008.

[8] H.-C. Jang and H.-C. Lee, “Efficient energy management to prolong wireless sensor network lifetime,” in *Proc. of 3rd IEEE/IFIP International Conference in Central Asia on Internet, ICI 2007.*, pp. 1-4, 2007.

[9] S. Oktug, A. Khalilov, and H. Tezcan, “3D Coverage Analysis under Heterogeneous Deployment Strategies in Wireless Sensor Networks,” in *Proc. of Fourth Advanced International Conference on Telecommunications, 2008. AICT '08*, pp. 199-204, 2008.

[10] S. C. Oh, C. H. Tan, F. W. Kong, Y. S. Tan, K. H. Ng, G. W. Ng, and K. Tai, “Multiobjective optimization of sensor network deployment by a genetic algorithm,” in *Proc. of IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, pp. 3917-3921, 2007.

[11] Y. Shang and H. Shi, “Coverage and energy tradeoff in density control on sensor networks,” in *Proc. of 11th International Conference on Parallel and Distributed Systems, 2005*, pp. 564-570, 2005.

[12] J. Wu and S. Yang, “Coverage issue in sensor networks with adjustable ranges,” in *Proc. of 2004 International Conference on Parallel Processing Workshops, 2004. ICPP 2004*, pp. 61-68, 2004.

[13] S. Zhang, Y. Liu, J. Pu, X. Zeng, and Z. Xiong, “An Enhanced Coverage Control Protocol for Wireless Sensor Networks,” in *Proc. of 42nd Hawaii International Conference on System Sciences, 2009. HICSS '09.*, pp. 1-7, 2009.

行政院國家科學委員會補助國內專家學者出席國際學術會議報告

年　　月　　日

附件三

報告人姓名	陳建宏	服務機構及職稱	中華大學 資訊工程系 助理教授
會議時間 地點	98 年 7 月 8 日至 98 年 7 月 12 日 加拿大蒙特婁	本會核定補助文號	
會議名稱	(中文) ACM 遺傳與演化式計算會議 2009 (英文) ACM Genetic and Evolutionary Computation Conference 2009		
發表論文題目	(中文) 3D 分化型感測網路佈建之多目標演化式最佳化 (英文) Multi-objective Evolutionary Optimization of 3D Differentiated Sensor Network Deployment		

一、參加會議經過

7月8日註冊報到，與會人士共計約千名來自學術界和工商業界之專業人士與學生。8日及9日分別依不同的主題舉辦專題討論和tutorial，10-12日分別依不同的主題在不同的會議室舉行口頭報告。9日晚間舉辦壁報論文發表暨歡迎會。會議中邀請到三位知名專家前來給予專題演講，其中最盛大的為11日早上遺傳演算法之父John Holland教授舉辦有關遺傳演算法之過去與未來。11日晚間所有專家學者被邀請至港口會場進行簡單的社交討論。12日中午發表論文。隨後前往美國拉斯維加斯參與 WORLDCOMP2009 發表論文。

二、與會心得

透過參加此次國際會議與來自各國的學者交流研究心得並且同時建立溝通管道。與本次會議中，與會的各國學者就其研究領域充份探討及意見交流，本人更於會後與美國麻省理工學院、伊利諾大學香檳校區、華盛頓大學聖路易士校區、英國諾丁罕大學、德國維爾茨堡大學、新加坡南洋理工大學、澳洲健保局專家、及台灣大學與交通大學等各國學者共同討論研究方向，並獲邀請前往參訪其所屬大學。會議所見所得對於未來個人學術研究開拓更寬廣的視野。個人能參與該會議備感榮幸。

三、建議

由此次的經驗，個人認為補助參與國際會議的政策對提昇台灣學者之研究能量和知名度有非常正面的助益，個人認為此一政策應持續推行，並且應鼓勵台灣學者教授積極攜帶博士生於寒暑期參訪各國學者且積極參與國際會議，不僅可增加台灣學者與碩博士研究生的國際觀和研究能力，更有助於促進台灣學術研究和學術交流之風氣。

四、攜回資料名稱及內容

1. 會議論文集光碟片 一片。