應用動態規劃式演化策略解決生產分配問題

Dynamic Programming Variant in Evolution Strategies for Production Allocation Problems

侯永昌 張應華
Young-Chang Hou Ying-Hua Chang
國立中央大學資管系
Department of Information Management, National Central University

摘 要

演化策略(evolution strategies)通常應用於目標變數為實數值的最佳化問題上，其主要是利用突變(mutation)的方式於解答空間中搜尋出最好的滿足解，利用類似的演化過程有許多組合最佳化的問題合適且成功地被解決，如二次指派問題(quadratic assignment problems)，工程設計最佳化問題(engineering design optimization problems)等等，但一般的演化計算技術應用於這些問題時，將會浪費很多的時間於處理在演化過程中獲得無效解的情形上，以致於造成演算法效率不彰。

本研究提出一種新的染色體編碼方式，稱為路徑編碼法(the path-encoding method)，其修改動態規劃的路徑搜尋概念於組合演化策略(combinatorial evolution strategies)上，以提高其演化效能，並以NP-hard 的生產分派問題(production allocation problems)來當成測試的實例，在實驗中對路徑編碼法、組合編碼法(combination-encoding method)、懲罰編碼法(penalty-encoding method)和整數規劃法(integer programming)等四種方法做一比較，結果顯示出本研究所提之路徑編碼法效果最好，其次是組合編碼法，再者是懲罰編碼法，最後是整數規劃法，其主要的原因是路徑編碼法可以有效地縮小搜尋解答空間的範圍，以加快收斂的速度。

關鍵字：演化策略、生產分派問題、動態規劃式演化策略、路徑編碼法、路徑突變法

Abstract

Evolution strategies are applied to optimize real-valued vectors of objective variables. These strategies rely primarily on mutation to explore the solution search space. Many combinatorial optimization problems such as quadratic assignment problems, engineering design optimization problems, and others can be successfully solved by analog evolution. This evolutionary algorithm wastes much time in managing invalid solutions and is typically less efficient.

This paper presents a new approach, called the path-encoding method, which modifies the path searching idea of dynamic programming for combinatorial evolution strategies to enhance the performance of evolutionary process. The NP-hard production allocation problem is used to evaluate the effectiveness of the approach. This experiment compares the proposed approach to the combination-encoding method, the penalty-encoding method and integer programming. The computed results show that the proposed approach is always feasible and outperforms the others because it narrows the solution search space.

Keywords: Evolution Strategies, Production Allocation Problems, DP Variant Evolution Strategies, Path Encoding Method, Path Mutation

1. Introduction

Using conventional algorithms to find an optimal solution for NP-hard problems, such as combinatorial optimization problems, is troublesome. Therefore, some other approaches such as approximation algorithms and evolutionary algorithms have been developed to find sufficiently good solutions to these NP-hard problems. In which, the evolutionary algorithms are probabilistic search algorithms that mimic biological evolution to produce better offspring solutions. A chromosome, which represents an instance of the population, encodes a solution to a given problem. This chromosome may be a string of bits, a string of real numbers or a tree-like string. Evolutionary algorithms assign fitness to every individual. According to the quality of the solution that each individual represents, fitter individuals in the population are more likely to survive to the next generation, and vice versa. Every generation must pass through all or some of the main operations - selection, crossover and mutation. This evolutionary cycle is repeated until a satisfactory solution is found. Nissen (1994), Yagiura et al (1996), Cai et al (1996), Li et al (1996), Chu et al (1997), Ahuja et al (2000)…and other researchers have proposed several evolutionary algorithm modifications, for specific problems. Evolutionary algorithms based on the principles of evolution and natural selections have successfully been applied to many complex problems in the areas of optimization, system identification, data mining and others (Biethahn et al. 1995).

Dynamic programming (DP) is a method that can solve several optimization problems. In most applications, it splits a complex problem into many simpler sub-problems, and then determines the optimum solution stage by stage. Dynamic programming restricts the acquisition of a good solution from the starting point to the goal point, obeying all constraints. It can yield valid solutions which slowly become better and better. The shortcoming is that dynamic programming typically wastes much time in finding all sub-solutions and generally expands in non-polynomial time to solve complex problems. Conventional dynamic programming is therefore inefficient to solve complex problems. Evolution strategies use the basic principles of replication, variation and selection from Darwin's evolutionary theory, but with some adjustment on solution's representation, selection scheme, strategic parameter adaptation and sequence of evolutionary operators. The most important characteristic of evolution strategies is that they explore the solution search space simultaneously from several points, as in parallel processing. Useful information derived from the objective function is used to find good solutions within the search space.

This study presents a variant of evolution strategies that can accelerate evolution. The proposed approach encodes the dynamic programming decision path into a chromosome to solve production allocation problems. The evolution of these paths can quickly determine a good solution. We provide an efficient resolution that proposes a new encoding method for evolution strategies. This new encoding method can greatly reduce the solution search space and more effectively find optimal solutions. A specialized mutation operator is proposed to enable this efficient encoding method to run smoothly. The production allocation problem, which is NP-hard, is used to evaluate the effectiveness of this approach. Using this test problem, the approach is compared to combination encoding (Hou et al. 2002), penalty encoding (Garavelli et al. 1996) and integer programming (Winston 1991). Computational experiments confirm that the new resolution model is always feasible and outperforms the others.

2. Evolution Strategies

Rechenberg (1973, 1994) and Schwefel (1977, 1995) developed evolution strategies to solve engineering optimization problems. The method resembles genetic algorithms but with some differences. These two kinds of evolutionary algorithm differ in the representation of the solution, selection scheme, strategic parameter adaptation and sequence of evolutionary operators. Evolution strategies usually use real-valued vectors to represent the solution, while genetic algorithms usually use binary vectors (Goldberg 1989). Evolution strategies rely primarily on mutation operations to explore the solution search space, while the dominant operator of evolutionary processes for genetic algorithms is crossover.

Evolution strategies randomly set an initial population and then calculate the fitness of individuals. The reproduction step can be implemented if the population cannot satisfy the objective function. In this step, λ children are created as an intermediate population by mutating μ parents, where the mutation operation uses normally distributed random variables. The fitness of all individuals in the population is then evaluated. The next step selects μ best individuals from intermediate population ((μ,λ)-selection) or selects μ best individuals from the set of parents and children ((μ+λ)-selection) as the new generation. This cycle is repeated until the termination criterion applies. Evolution strategies differ from traditional search and optimization techniques in that the former simultaneously explore the solution search space from several points. Information derived from the objective function (fitness) is required to guide the search for good solutions within the search space. Moreover, evolution strategies rapidly complete optimization because the random distribution of new trials concentrates the computational effort on solutions that were previously proven successful, reducing the computational effort (Cai et al. 1996; Nissen 1994).

General evolution strategies are typically applied to real-valued vectors of the objective variables to be optimized. Many combinatorial optimization problems can be solved through analog evolution, which is the combinatorial variant of evolution strategies (Chang 2000; Nissen 1994). Solving the different combinatorial optimization problems usually depends on a problem-specific encoding method and self-adaptive correlated mutations to guide the search process more efficiently.

3. Production Allocation Problems

Production allocation problems concern the assignment of global demand for a product to a multinational company characterized by subsidiaries located in different geographical areas (Garavelli et al. 1996). They involve allocating plant output among many markets to minimize the costs to the multinational company subject to capacity constraints and market demand (Bhatnagar et al. 1993; Lootsma 1994). Figure 1 represents such a production allocation problem.

The left side nodes represent the plants (i) that produce products to meet the demands of all of the markets (j) on the right side. The manufacturing costs per unit production in plant i are expressed by m_{ij}. t_{ij} represents the transportation costs per unit. d_{ij} represents the import duties and taxes per unit for a product shipped from plant i to market j. q_{ij} represents the quantity of product produced by plant i to support market j.C_i is the maximal production capacity of plant i, and D_j denotes the minimum demand of market j. Hence, the costs of plant i to produce a unit of product to supply to market j can be written as,

$$Cost_{ij} = (m_{ij} + t_{ij} + d_{ij}) \times q_{ij}$$

For simplicity, the plants are considered to produce a single product. The production allocation problem can thus be modeled as,

$$min\ \text{Cost} = min\left\{\sum_{i,j} Cost_{ij}\right\}$$

$$subject\ to \begin{cases} \sum_{j} q_{ij} \le C_i \text{....................(1)} \\ \sum_{i} q_{ij} \ge D_j \text{...................(2)} \\ q_{ij} \ge 0,\ and\ q_{ij} \in Z \end{cases}$$

where inequality (1) states that the total quantity that each plant supplies to each market must be less than the production capacity of each plant. Inequity (2) specifies that the demand of each market must be met by the total production from all plants. The variable q_{ij} is a discrete nonnegative integer variable.

This problem is a combinatorial optimization problem and can be recognized as an NP-hard problem (Garavelli et al. 1996), such that solving this problem within polynomial time using general algorithms is impossible. Using iterative and random search methods, which involve parallel processing, an evolutionary process has a powerful solution searching capability. These features characterize the general-purpose search and optimization techniques that are applicable to several difficult problems (Nissen 1994).

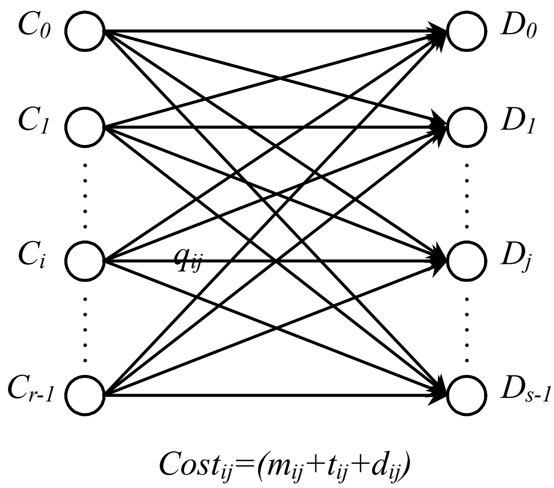


Figure 1. Production allocation model

This study used evolution strategies to solve the production allocation problem. An efficient new encoding method, called path encoding, is proposed along with a specialized mutation operation, called path mutation. The following section introduces the DP variant evolution strategies encoding method with self-adaptive correlated mutations, to guide more efficiently the search process in combinatorial optimization problems.

4. DP Variant Evolution Strategies to Solve Production Allocation Problems

This paper presents a new encoding method and a mutation operator that is abstracted from the benefits of dynamic programming to perform the evolutionary procedure of evolution strategies, called DP variant evolution strategies. This encoding method and the corresponding mutation operator can restrict the evolutionary process to check only valid solutions. This solution procedure involves parallel processing, which is the advantage of evolution strategies. Incorporating the idea that a dynamic programming decision path must obey all constraints of the problem and obtain only valid solution from all evolutionary processes, evolution strategies are become more efficient when solving complex problems. The DP variant evolution strategies proceed as follows.

DP variant Evolution Strategies( )
{ Applying **path encoding method** to generate initial population
Evaluate the initial individuals
Repeat
{ Reproduction (**path mutation**)
Selection }
Until (termination criterion holds) }

In DP variant evolution strategies, path encoding is applied to the population to generate valid individuals. A problem-specific mutation operator is proposed to match the evolutionary procedure to ensure that the intermediate chromosomes are always valid.

Figure 2 shows the approach. In the initial stage, DP variant evolution strategies generate *p* (*p* is the population size) valid individuals. After the problem-specific mutation operator is applied to individual 1, 2, …, *i*,…, *p* and the selection operator is implemented, the second generation acquires new individuals 1´, 2´,…, *i*´,…, *p*´. Using the analog evolutionary process, a better solution is rapidly obtained after a few generations. This new approach is suited to combinatorial problems.

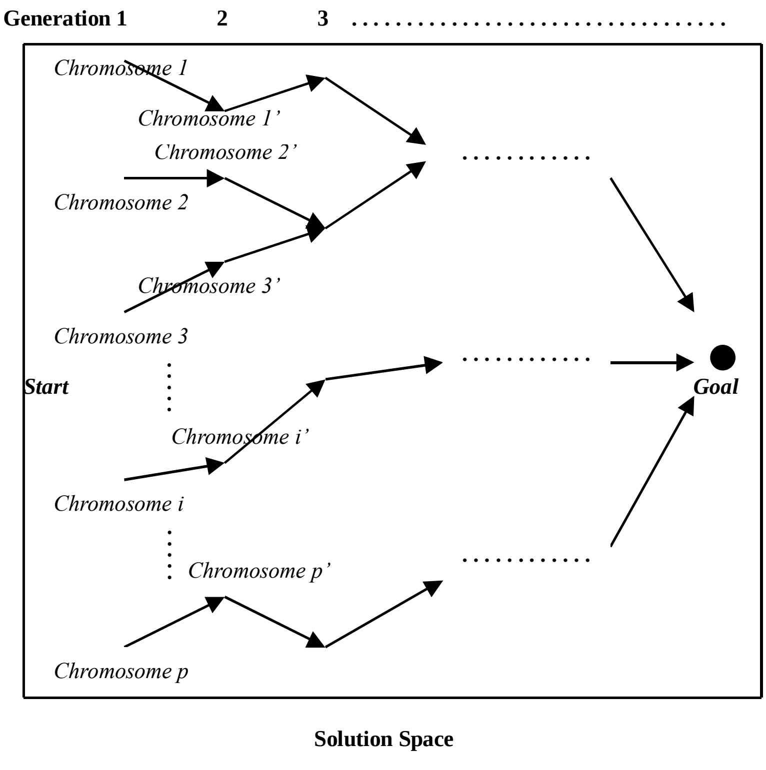


Figure 2. The evolution process of DP Variant Evolution Strategies

Initialization and Path Encoding Method

The production allocation model allocates output from each of the international company's plants to the different markets. Consider s markets and r plants. The new path encoding method is developed for this problem to enhance the evolution performance and yield better offspring for DP variant evolution strategies. Assuming s markets, M_0, M_1, …, M_j, …, M_{s-1}, a corresponding market demand of D_0, D_1, …, D_j, …, D_{s-1} and r plants, P_0, P_1, …, P_i, …, P_{r-1}, the corresponding production capacity of the plants is C_0, C_1, …, C_i, …, C_{r-1}. Figure 3 depicts how the encoding of the chromosome.

Each segment j (j=0 ~ s-1) represents the demand in a market. The integer number in each subsegment, q_{ij}, is the quantity or quantity percentage of products supplied by plant i (i=0~r-1) to market j. If the demands are quite large, then q_{ij}'s are used to represent the percentage supplied. This will restrict the length of the chromosome to a reasonable size. Markets' demands are sequentially assigned by plants to prevent an invalid chromosome from being obtained, which occur when the total products supplied by plant i exceeds the plants' production capacity, or when the demand in some markets cannot be satisfied. q_{ij} (i=0 ~ r-1) of segment j continues to be processed until segment j is completely assigned. Then, segment j+1 mod s begins to be processed. This assignment process can be started from any markets. Figure 4 describes the above procedure.

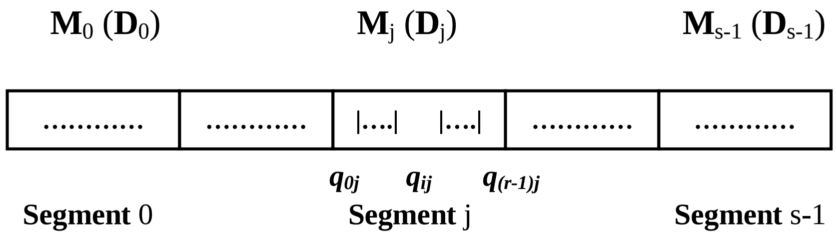


Figure 3. Chromosome in the production allocation problem

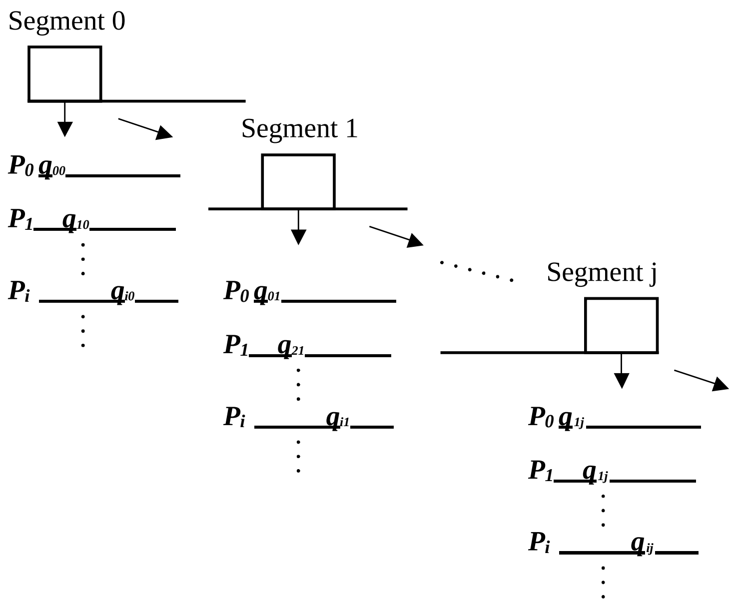


Figure 4. DP variant Evolution Strategies encoding procedure

There are some restrictions on the assignment of q_{ij} for plant i to market j. Because plant i can't supply more than its capacity C_i, in the mean time, and the demand of market j should be satisfied. The value of q_{ij} should lie within a condition-specific region. The upper bound is the minimum value of surplus capacity of plant i or the remaining demand of market j. The lower bound is the difference between the remaining demand of market j and the remaining product capacity of other plants. They are expressed as follows.

Upper bound:

$$min\left\{C_i - \sum_{x=1}^{j-1} q_{ix}, D_j - \sum_{x=1}^{i-1} q_{xj}\right\} \ldots(3)$$

Lower bound:

$$Max\left\{0, D_j - \sum_{k=1}^{i-1} q_{kj} - \sum_{k=i+1}^{t} (C_k - \sum_{l=1}^{j-1} q_{kl})\right\} ..(4)$$

in which, $C_i - \sum_{x=1}^{j-1} q_{ix}$ represents the present surplus capacity of plant i, and $D_j - \sum_{x=1}^{i-1} q_{xj}$ is the remaining demand of market j, and $D_j - \sum_{k=1}^{i-1} q_{kj} - \sum_{k=i+1}^{t} (C_k - \sum_{l=1}^{j-1} q_{kl})$ is the remaining demand of market j minus the maximum remaining production from other plants. When q_{ij} is between these two limited values, inequalities (1) and (2) of the production allocation model are always satisfied by the chromosomes of the evolution strategies.

This method can reduce the solution search space, allowing effective solutions to be more rapidly obtained in the evolutionary process. Applying the upper and lower bounds can generate chromosomes when initializing the population, and ensures that the chromosomes do not violate inequalities (1) and (2) of the production allocation model. An example is presented below:

Suppose three plants (P_0, P_1, P_2), meet the demand of two markets (M_0, M_1). The corresponding production capacity of these three plants is C_0=50, C_1=40, C_2=50. The demand of the two markets is D_0=70 and D_1=60. Figure 5-(1) depicts a valid chromosome since each q_{ij} falls between its upper and lower bounds. The total quantity of all plants support to these two markets can meet their demand, that is $\sum_i q_{i0} = D_0 (= 70)$ and $\sum_i q_{i1} = D_1 (= 60)$. Besides, the production quantity of each plant doesn't violate its production capacity, that is $\sum_j q_{0j} \le C_0 (= 50)$, $\sum_j q_{1j} \le C_1 (= 40)$ and $\sum_j q_{2j} \le C_2 (= 50)$. Figure 5-(2) shows an invalid chromosome. Although all plants can satisfy the demand of these two markets, $q_{10}+q_{11}$ (=50) exceeds the production capacity of plant 1 (C_1=40).

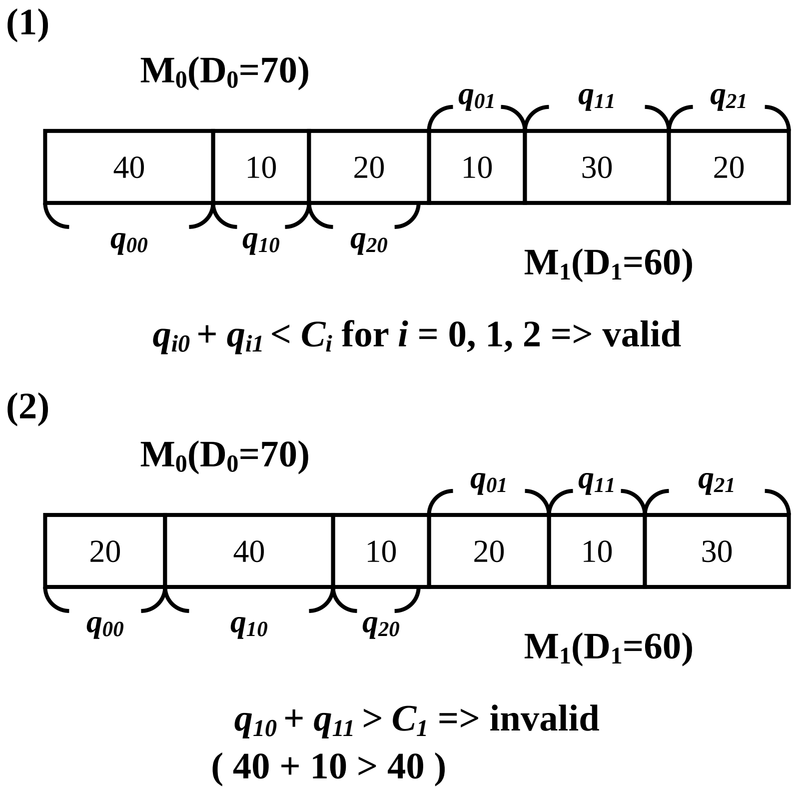


Figure 5. Examples of valid and invalid chromosomes

Evaluation

Once q_{ij} satisfies equation 3 and 4, the path encoding method directly codes inequalities (1) and (2) of the production allocation model into the chromosome. Consequently, the chromosome will satisfies the required constraints, i.e. the quantity provided by each plant to all markets is less than that plant's production capacity, and the demand of each market can be satisfied. The total cost to each plant of supplying products to a market can be used to evaluate of each chromosome. The chromosome has a higher probability to survive in the next generation when it has a higher fitness. This approach can improve the performance of the evolution strategies in searching for effective solutions, because all constraints are encoded into individuals.

Selection

Two general methods exist for maintaining a population of solutions in the evolution strategies - (μ+λ)-selection and (μ,λ)-selection. The (μ+λ)-selection involves μ parents that generate λ offspring and puts μ+λ individuals in competition to survive. Only the best μ individuals survive to the subsequent generation. The (μ,λ)-selection excludes the parents of each generation, such that only the children compete for survival. The best μ children remain to form the subsequent generation. Generally, (μ,λ)-selection is preferable because it allows for a temporal deterioration of the population's best solution (Schwefel 1995). This deterioration may be required to overcome a local optimum and prevent premature convergence. Hence, this study uses (μ,λ)-selection.

Path Mutation

The general mutation method for evolution strategies is inadequate for the combination problem because it may yield invalid results. Therefore, a suitable mutation operator is proposed herein for production allocation problem. This new mutation operator first arbitrarily selects a segment, say j, which represents the demand of market j and then chooses any subsegment, say i. The number of q_{ij} in segment j represents the quantity or quantity percentage of a product provided by plant i to meet demand in market j. After a subsegment is randomly selected, a few number (δ) are added to this subsegment, and the same number (δ) are deleted from the adjacent, i+1, subsegment in order to satisfy the demand of market j. And the small random number, δ, is normally distributed. If the inequalities (1) and (2) are still met following this mutation operation, the mutation process is completed. However, this action may cause one of these two subsegments to violate the production capacity of the corresponding plants. If such a violation occurs, the production value of the corresponding two plants in the next segment, j+1, must be deleted for an equal number (δ) in subsegment i and added for an equal number (δ) in subsegment i+1, until chromosome is valid. After mutation, the new chromosome is generated and continues to satisfy to the constraints.

Figure 6 depicts a valid chromosome that underwent mutation. Segment j may undergo mutation by adding two units' products in the P_i block and deleting two units' products from the P_{i+1} block for a chromosome. When the corresponding plants i and i+1 does not violate the production capacity of the plants after the blocks P_i and P_{i+1} in segment j are mutated, segment j+1 is not altered. The new chromosome may become chromosome´. Otherwise, the new chromosome resembles chromosome´´, in which the part that is blocked in segment j+1 may delete two units' products from P_i and add two units' products to P_{i+1}.

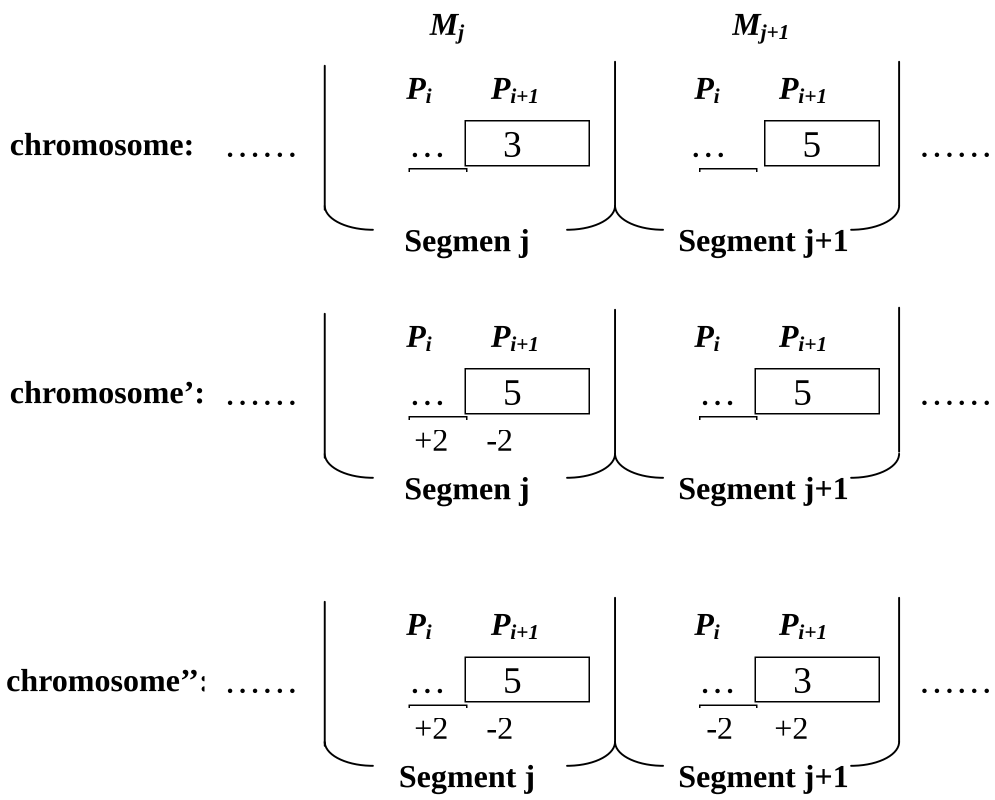


Figure 6. Chromosome mutation

5. Illustrative Examples

Three highly complex scenarios were built and tested using the following four methods - the new path encoding method, combination encoding (Hou et al. 2002), penalty encoding (Garavelli et al. 1996) and integer programming (Winston 1991), to show that the new path encoding method is highly efficient when applied to combinatorial problems. These scenarios were modified from Garavelli et al (1996), and involve characterized by demand from three, four and five markets for a product supplied from five, six and seven plants of a global manufacturing company. The three cases are (three markets, five plants), (four markets, six plants) and (five markets, seven plants). Table 1 shows a simple example of the (three markets, five plants) case. The other two scenarios are similar.

P_i (i=0 ~ 4) represents the five plants. The quantity of product produced at each plant is given. The corresponding value C_{ijq} (j=0 ~ 2) represents the cost to plant i of producing q units of the product to meet the demand of market j. The demand of the three markets is 70, 60 and 80 respectively. The production capacities of the five plants are 50, 40, 50, 45 and 50. Each plant must produce q_{ij} number products to satisfy the demand of all markets, while minimizing the cost of the company. The computational complexity of this scenario is 3.24559×10^{33}, and that of the other two scenarios is 5.9368×10^{40} and 3.65631×10^{47}, making them high complexity problems.

Table 1. Configuration for the test scenarios

Cost	Demand of Market 0					Demand of Market 1					Demand of Market 2				
Units	P_0	P_1	P_2	P_3	P_4			P_i					P_i		
0	xx	xx	xx	xx	xx			xx					xx		
1	xx	xx	xx	xx	xx	....		xx	....		....		xx	....	
2	xx	xx	xx	xx	xx			xx					xx		
3	xx	xx	xx	xx	xx			xx					xx		
⋮			⋮					⋮					⋮		
⋮			⋮					⋮					⋮		
q			C_{20q}					C_{i1q}					C_{i2q}		
⋮			⋮					⋮					⋮		
⋮			⋮					⋮					⋮		

The new path encoding method encodes two constraints into the chromosome. These are the demand of markets and the production capacity of plant. The combination encoding method encodes one constraint (market demand) into the chromosome. The second constraint is limited by the penalty function. The penalty encoding method must exploit the penalty function to cause the evolutionary algorithm to meet these two constraints and yield good solutions.

These algorithms were coded in C and run on an IBM compatible PC. This new path encoding method can encode all constraints into the chromosomes such that the search space is smaller than that of other encoding methods that can encode only one constraint into the chromosome, or use only the penalty function to guide the evolutionary algorithms. Figure 7 shows the search space of these three encoding methods.

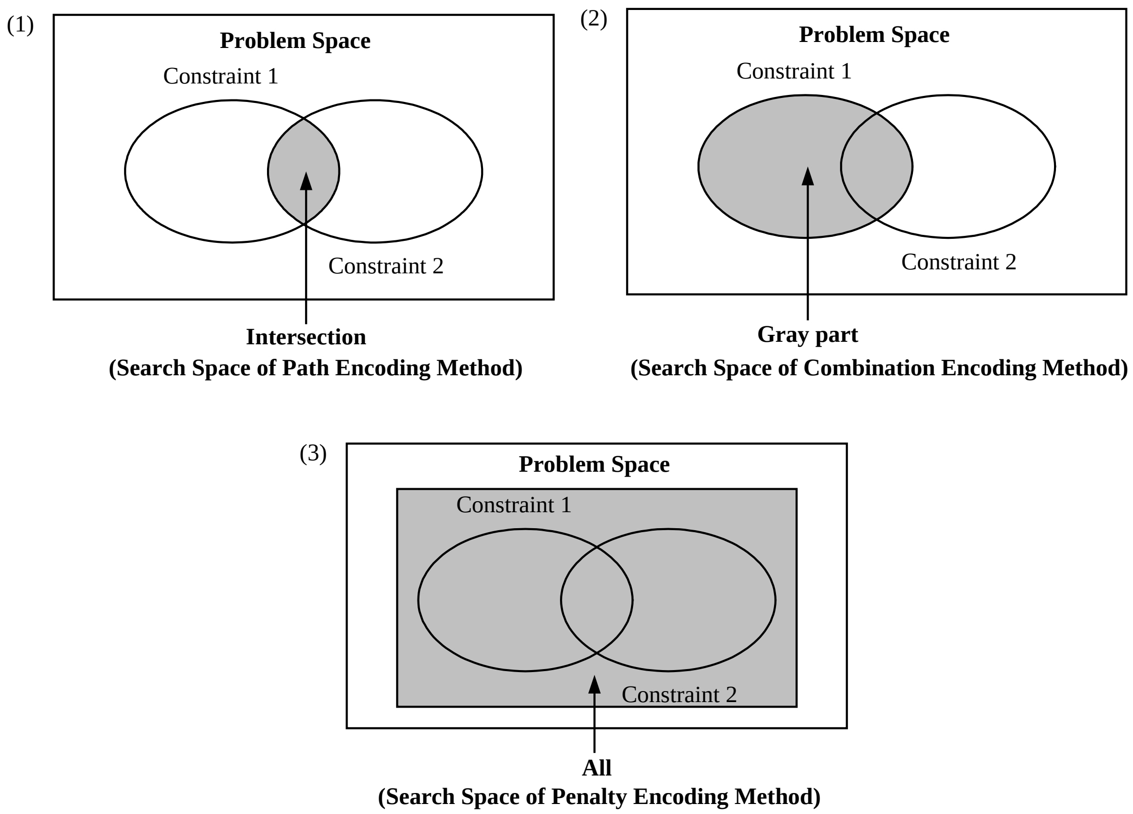


Figure 7. Search space of three encoding

Figure 7-(a) reveals that the intersection of constraints (1) and (2) specifies the search space of the path encoding method. Figure 7-(b) shows that the shaded region of constraints (1) is the search space of the combination encoding method. The whole shaded part in Fig. 7-(c) is the search space of the penalty encoding method. This figure reveals that the path encoding method may search in a smaller solution search space than other methods. As the constraint number is over than two, we can take advantage of penalty function to guide the evolution process to get sufficient good solutions.

Table 2 compares the results achieved by the three encoding methods and integer programming applied to the test problem. The first column gives the generation that leads these three encoding methods and integer programming to converge to good solutions. Ten trials were performed for each experiment. The (*s* markets, *r* plants) subcolumn shows the fitness (total cost) of the best chromosomes from the experiments. If the value of the (*s* markets, *r* plants) sub-column is invalid, then the evolutionary algorithm cannot yield a feasible solution until the corresponding number of generations has reached. A value is valid if the evolutionary algorithm begins to find feasible solutions.

Table 2. Computational results of the test scenario

Cost / **Generations**	**Path Encoding**			**Combination Encoding**			**Penalty Encoding**		
	(3M, 5P)	(4M, 6P)	(5M, 7P)	(3M, 5P)	(4M, 6P)	(5M, 7P)	(3M, 5P)	(4M, 6P)	(5M, 7P)
120	Valid	Valid	Valid	Valid	Invalid	Invalid	Invalid	Invalid	Invalid
200	219								
350		269			Valid				
500			310			Valid			
3000							Valid		
8000				219				Valid	Valid
17000					269				
30000						310			
40000							223		
90000								277	
150000									319

Integer Programming		
(3M, 5P)	(4M, 6P)	(5M, 7P)
Integer Programming only can get feasible solutions but not good solutions		
feasible solution	feasible solution	feasible solution
252	309	363

For the (three markets, five plants) case, Table 2 shows that the new path encoding method significantly outperforms the other two encoding methods and integer programming, requiring just 200 generations and finding good solutions (total cost=219). This performance is good because the solutions of the new encoding method in the search space are always feasible. The result matches the assumption that the suitable path mutation operators do not generate infeasible chromosomes such that good solutions are obtained more quickly. The combination encoding method must run for around 120 generations to gain valid solutions and reach 8000 generations to yield good solutions (total cost=219). This many generations are required because the combination encoding method encodes only a single constraint (market demand) into the chromosomes. Before 120 generations, the evolutionary algorithm tends to yield valid chromosomes with the help of the penalty function. A valid chromosome may be invalidated when the crossover and mutation operators are implemented. This situation may delay obtaining feasible solutions by evolutionary algorithms. The penalty encoding method takes even more time than the combination encoding method to obtain valid chromosomes up approximately 3000 generations and also requires more effort to find good solutions. It is also likely to result in local optimization (total cost=223). Good solutions are slowly obtained because this encoding method merely exploits the penalty function to cause the evolutionary algorithms not to violate these two constraints. This test scenario is highly complex so the penalty encoding may fail due to local optima. Therefore, these two encoding methods are always less likely to find good solutions than the path encoding method and require more time to converge. The LINGO package was used to solve these three scenarios by integer programming. For the (three markets, five plants) case, only LINGO can find feasible solutions (total cost=252) because the scenarios are all highly complex. The interpretations of the other two scenarios are as above.

6. Conclusions

The effectiveness of an algorithm can be roughly determined by the size of the solution search space (Hou et al. 2002). A smaller search space corresponds to a more effective algorithm. Hence, one of the most important ways to improve the performance of an algorithm is to narrow the solution search space explored by the algorithm. The general evolution strategies applied to combinatorial problems waste much time in managing invalid solutions during evolution. This study proposed dynamic programming variant evolution strategies, employing the path encoding method and mutation operator, to solve this problem.

The concept of upper and lower bound is used to restrict the valid decision path that satisfies the constraints and thus yields a good solution from the initial to final states of decision process. A combinatorial problem, the production allocation problem, was used as a benchmark to test the feasibility and effectiveness of this new approach. Computational experiments tested the new method, combination encoding, penalty encoding and integer programming. The experimental results prove that our new path encoding method and mutation operator perform better than other methods in finding good solutions. Our new approach greatly narrows the search space, accelerating convergence to a solution. In the future, this new approach will be applied to other combinatorial problems.

Reference

Ahuja, R.K., Orlin, J.B. and Tiwari, A. "A Greedy Genetic Algorithm for The Quadratic Assignment Problem." *Computers & Operations Research* (27), 2000, pp.917-934.

Bhatnagar, R., Chandra, P. and Goyal, S.K. "Models for Multi-plant Coordination." *European Journal of Operational Research* (67:2), 1993, pp.141-160.

Biethahn, J. and Nissen, V. *Evolutionary Algorithms in Management Applications,* Springer, 1995.

Cai, J. and Thierauf, G. "Evolution Strategies for Solving Discrete Optimozation Problems." *Advances in Engineering Software* (25), 1996, pp.177-183.

Chang, Y.H. "A New Combination Encoding of Genetic Algorithms for Timetabling Problems." In: K.K. Lai, O. Katai, M. Gen and B. Liu., Proceedings of the Second Asia-Pacific Conference on Genetic Algorithms and Applications, 2000, pp.98-107.

Chu, P.C. and Beasley, J.E. "A Genetic Algorithm for The Generalized Assignment Problem." *Computers Operations Research* (24:1), 1997, pp.17-23.

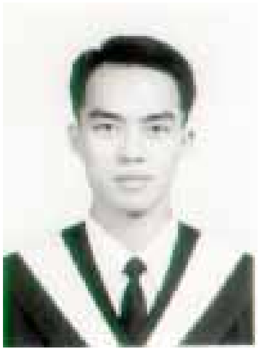
Garavelli, A.C., Okogbaa, O.G. and Violante, N. "Global Manufacturing Systems: A Model Supported by Genetic Algorithms to Optimize Production Planning." *Computers & Industrial Engineering* (31), 1996, pp.193-196.

Goldberg, D.E. *Genetic Algorithm in search,*

optimization and machine learning. Addison Wesley, 1989.
Hou, Y.C. and Chang, Y.H. "A New Efficient Encoding Mode of Genetic Algorithms for the Generalized Plant Allocation Problem." accepted by *Journal of Information Science and Engineering*, 2003.
Li, D.C., Lin, H.K. and Torng, K.Y. "A Strategy for Evolution of Algorithms to Increase The Computational Effectiveness of NP-Hard Scheduling Problems." *European Journal of Operational Research* (88), 1996, pp.404-412.
Lootsma, F.A. "Alternative Optimization Strategies for Large-Scale Production-Allocation Problems." *European Journal of Operational Research* (75:1), 1994, pp.13-40.
Nissen, V. "Solving The Quadratic Assignment Problem With Clues From Native." *IEEE Transactions on Neural Networks* (5:1), 1994, pp.66-72.
Rechenberg, I. *Evolutionsstrategie. Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Frommann-Holzboog, Stuttgart, 1973.
Rechenberg, I. *Evolutionsstrategie'94.* Frommann-Holzboog, Stuttgart, 1994.
Schwefel, H.P. *Evolution and Optimum Seeking.* John Wiley, New York, 1995.
Schwefel, H.P. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie.* Birkhauser, Basel, 1977.
Winston, W.L. *Operations Research-Applications and Algorithms*, Pws-Kent, Boston, 1991.
Yagiura, M. and Ibaraki, T. "The Use of Dynamic Programming In Genetic Algorithms for Permutation Problems." *European Journal of Operational Research* (92), 1996, pp.387-401.

作者簡介：

國立交通大學資訊工程研究所博士。現任國立中央大學資訊管理系副教授，兼學生事務處諮商輔導中心組長。研究領域爲資訊隱藏、浮水印技術與視覺密碼、模糊理論、軟體工程、演算法則。



國立中央大學資訊管理研究所博士班研究生。研究領域爲人工智慧、柔性計算技術、作業研究、決策分析、智慧型資訊系統。