SOLVING THE SINGLE MACHINE SCHEDULING PROBLEM WITH SEQUENCE-DEPENDENT SET-UP TIMES AS A TSP VIA EVOLUTIONARY ALGORITHMS

Minetti Gabriela, Alfonso Hugo
Proyecto UNLPAM-09/F015[1]
Departamento de Informática - Facultad de Ingeniería
Universidad Nacional de La Pampa
Calle 110 esq. 9 – Local 14
6360 General Pico – La Pampa – Rep. Argentina
e-mail: {minettig, alfonsoh }@ing.unlpam.edu.ar
Phone: +54 02302 422780/422372, Ext. 6302

Gallard Raúl
Proyecto UNSL-338403[2]
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950 - Local 106
5700 - San Luis - Rep. Argentina
e-mail: rgallard@unsl.edu.ar
Phone: +54 2652 420823 Fax: +54 2652 430224

SUMMARY

Scheduling is an active area of research in applied artificial intelligence. Scheduling problems typically comprise several concurrent (and often conflicting) goals, and several resources which may be allocated in order to satisfy these goals. In many cases, the combination of goals and resources results in an exponentially growing problem space. As an immediate result, no deterministic method exists for solving those problems in polynomial time. Such problems are called NP-complete problems, with respect to the exponential time and memory requirements necessary to reach optimal solutions. Approaches to scheduling have been varied and creative. Examples of the different underlying scheduling techniques are local search, simulated annealing, constraint satisfaction, evolutionary computation, among others. The problem is choosing the appropriate technique for a specific type of scheduling application.

Scheduling problems are often complicated by a large number of constraints relating activities to each other. The mathematical evaluation function is highly complex and the search space is too large with many local maximum [6]. The set-up times are one of the most common complications in scheduling problems. The set-up times are defined [4] as the time intervals between the end of a job processing and the beginning of the next job. In this time interval no jobs can be processed in the machine. For single machine [5] scheduling problems without sequence-dependent set-up times, the makespan is independent of the sequence and equal to the sum of the processing times. When there are sequence-dependent set-up times, the makespan depends on the schedule.

The single-machine scheduling problem with sequence-dependent set-up times can be described as follows [1]: a set of jobs J_1 .... J_n have to be scheduled on one machine; each job J_j , $j \in N = \{1,........,n\}$ has a release date r_j, a processing time p_j, and a delivery time d_j. Each job cannot be processed before its release time.

The TSP is one of the most widely studied and challenging NP-hard combinatorial optimization problems. Here, a travelling salesman wants to visit each of n cities starting and ending at a designated city 1. He visits no other city twice. Let $c_{ij} \geq 0$ the cost (or distance) between city i and city j. When there is no direct connection between them, we assign $c_{ij} = \infty$. The optimization problem is to find a minimum cost (shortest) tour.

[1] The Research Group is supported by the Universidad Nacional de La Pampa.
[2] The Research Group is supported by the Universidad Nacional de San Luis and the ANPCYT (National Agency to Promote Science and Technology).
http://www-pr.unsl.edu.ar/proyecto338403//home_page.html

Hence scheduling problems are prime candidates for the application of Evolutionary Algorithms. These algorithms have been used successfully in scheduling and travelling salesman problems (see for example [1,2,3]). The TSP is NP-hard, which probably means that any algorithm which computes an exact solution of the TSP requires an amount of computation time which is exponential in the size of the problem.

The similarities between the TSP and the single machine scheduling problem with sequence-dependent set-up times are the following [6]: the jobs can be seen as cities, while the sequence-dependent set-up times can be compared with the distance between the cities. Here the main goal is to determine a job sequence (a schedule) with the minimal makespan. In this problem the cost to go from job J_1 to job J_2 can be different to the cost of going from J_2 to J_1. Consequently, the asymmetric TSP variant is necessary for solving this problem.

Michalewicz et al. and Tao et al. [7, 8] proposed an evolutionary algorithm based on a special inver-over operator, which incorporates the knowledge taken from other individual in the population. The operator is a mixture of inversion and recombination. The inversion is applied to a part of a single individual, but the selection of a segment to be inverted depends on other individual in the population. In figure 1 the scheme of this algorithm is showed. The method provided optimal or near-optimal solution very fast and outperformed other evolutionary operators proposed in the past for the TSP (PMX, OX, CX, ER, etc).

```
random initialization of the population P
while (not satisfied termination-condition) do
        {
for each individual Si ∈ P do
{
        S′ = Si
 select (randomly) a city c from S′
 repeat
 {
          if (rand( ) ≤ p)
            select the city c′ from the remaining cities in S′
          else
          {
           select (randomly) an individual I from P
           assign to c′ the "next" city to the city c in the selected individual I.
          }
          if (the next city or the previous city of city c in S′ is c’)
         exit from repeat loop
            inverse the section from the next city of city c to the city c′ in S′
             c = c′
           }
              if (eval(S′) ≤ eval(Si))
                   Si = S′
        }
}
```

Fig. 1. The Evolutionary algorithm using the inver-over

In EAs the common approach to crossover is to operate once on each mating pair after selection. Such procedure is known as the SCPC (Single Crossover Per Couple) approach. But in nature when the mating process is carried out, crossover is applied many times and the consequence is a multiple and variable number of offspring. *Multiple crossover per couple* (MCPC) [10] is a novel crossover method. For each mating pair MCPC allows a variable number of children. A posterior extension of multirecombination, *multiple crossovers on multiple parents* (MCMP) allows multiple recombination of multiple parents, expecting that exploitation and exploration of the problem space be adequately balanced [11]. Both multirecombinative approaches are part of a wider family of EAs: those including multiplicity of contributors and operators to exchange genetic material. They are called *Multiplicity Feature Evolutionary Algorithms* (MFEA).

With a new perspective due to the nature of the operator, MFEAs can also be devised for the inver-over operator. The idea is to continue applying on the same current S' individual, for a predetermined number n_1 of times, the inver-over operation expecting to find better solutions. In other words, when comparing the evaluations of $S'$and S_i (original individual), if S' does not improves S_i then the loop for the inver-over operation is repeated (S_i again undergoes inver-over), for a maximum number n_1 of times. In this case, we facilitate that many other individuals of the population compete as donors until eventually a better offspring is created.

We study the application of those operators for symmetric and asymmetric TSP; solving in this way, the symmetric TSP and the scheduling problem with sequence-dependent set-up times, respectively.

CONCLUSIONS AND FUTURE WORKS

These multirecombinative variants of the inver-over approach, were applied to the Travelling Salesman Problem and the single machine scheduling problem with sequence-dependent set-up times. These variants were contrasted against the original method. Feasible offspring are generated by means of the inver-over operator. At the light of these results we can conclude that: regarding quality of results and speed to find near optimal solutions all methods including multirecombination outperform the original method, concerning all the considered performance variables, their values are improved as long as the number n_1 of inver-over operations is increased. Particularly, the optimum was found when one of the considered instances (br17) [9] was used for the scheduling problem with sequence-dependent set-up times, under both approaches.

To determine potentials and limitations of the evolutionary approach on scheduling problems further research include the study of the effect of more complex instances and other kinds of scheduling problem.

BIBLIOGRAPHY

1 Bagchi, S., Uckun, S., Miyabe, Y., and Kawamura, K., *Exploring Problem-Specific Recombination Operators for Job Shop Scheduling*, in Proceedings of the Fourth International Conference on Genetic Algorithms and Their Applications, San Diego, CA, pp10-17,1991

2 Cleveland G., and Smith, S., *Using Genetic Algorithms to Schedule Flow Shop Releases*, in Proceedings of the Third International Conference on Genetic Algorithms and Their Applications, Arlington, Va, pp.160-169, 1989.

3 Davis, L., *Job Shop Scheduling with Genetic Algorithms*, in Proceedings of the First International Conference on Genetic Algorithms and Their Applications, 1985.

4 Noivo J., Ramalhinho-Lourenco H., *Solving two production scheduling problems with sequence-dependent set-up times*, Journal of Economic Literature Classification: C61, M11, L60.

5 Pinedo, M., *Scheduling Theory, Algorithms, and Systems*, Columbia University, Prentice Hall, Englewood Cliffs, New Jersey 07632, 1995

6 In, L., Sikora, R., Shaw, M., *Joint Lot Sizing and Sequencing with Genetic Algorithms for Scheduling: Evolving the Chromosome Structure*, Beckman Institute for Advanced Science and Technology and Department of Business Administration University of Illinois at Urbana-Champaign, 1993.

7 Michalewicz Z., Esquivel S., Gallard R., Michalewicz M., Tao G, Trojanowski, K., The Spirit of Evolutionary Algorithms, special issue on Evolutionary Computing of the Journal of Computing and Information Technology, University Computing Centre, Zagreb, Croatia, pp.1-18, 1999.

8 Minetti G., Gallard R., Alfonso H., Inver-Over variants for de Euclidean Travelling Salesman Problem, Proceedings of the Second International ICSC Symposium on Engineering Of Intelligent Systems, EIS'2000, Scotland, pp. 458-463, 2000.

9 Reinelt, G., *TSPLIB 95*, Universität Heidelberg, Institut für Angewandte Mathematik, 1995.

10 Esquivel S., Leiva A., Gallard R., - *Multiple Crossover per Couple in Genetic Algorithms*, Proceedings of the Fourth IEEE Conference on Evolutionary Computation (ICEC'97), pp 103-106, ISBN 0-7803-3949-5, Indianapolis, USA, April 1997.

11 Esquivel S., Leiva H.,.Gallard R., *Multiple crossovers between multiple parents to improve search in evolutionary algorithms*, Proceedings of the 1999 Congress on Evolutionary Computation (IEEE). Washington DC, pp 1589-1594.