Diseño Inteligente de Redes Inalámbricas de Comunicación
Conserjería de Innovación, Ciencia y Empresa
P07-TIC-03044
`http://diricom.lcc.uma.es`

Málaga, 2010

Informe Ejecutivo

TÍTULO:	ACP-2.0: Resolución del problema ACP con metaheurísticas Grid
RESUMEN:	Este entregable aborda la resolución de una instancia de gran dimensión del problema ACP utilizando un algoritmo paralelo para sistemas de computación Grid.

OBJETIVOS:

1. Presentar una versión grid del algoritmo NSGA-II: assNSGA-II.
2. Obtener reducciones en significativas en los tiempos de ejecución.

CONCLUSIONES:

1. assNSGA-II es capaz de reducir de días a horas los tiempo de cómputo para instancias ACP de gran dimensión.
2. El algoritmo permite mejorar numéricamente la versión secuencial.

RELACIÓN CON ENTREGABLES:

PRE: ACP-1.0

CO: —

Málaga, 2010

Executive Summary

TITLE:	ACP-2.0: Solving the ACP problem with Grid-enabled metaheuristics

ABSTRACT: This deliverable addresses a large instance of the ACP problem by using a grid-enabled version of the NSGA-II algorithm.

GOALS:

1. assNSGA-II: a grid-enabled version of NSGA-II.
2. Reach significative reductions in the computational times.

CONCLUSIONS:

1. assNSGA-II is able to reduce the computational time from days to hours.
2. The numerical efficiency of assNSGA-II is also improved with respect its sequential counterpart.

RELATION WITH DELIVERABLES:

PRE: ACP-1.0

CO: —

Solving the ACP problem with Grid-enabled metaheuristics

DIRICOM

2010

1. Introduction

Planning and managing a cellular phone system make engineers face many challenging optimization problems ([9]). Assuming that the business planning activities are already completed, i.e., choosing customer segments, network technology to be used, etc., one of the most significant technical optimization problem is the radio network planning ([6]), also known as the *Automatic Cell Planning* (ACP) problem, the *network dimensioning* problem, or the *capacity planning* problem. Indeed, the foundation of a well-performing cellular network is the basic radio platform since it is the part of the network which is closer to mobile users. Also, it has a clear benefits for the operators since they reduce the infrastructure costs and, at the same time, increase revenue and user satisfaction.

In the initial deployment of a cellular network, the ACP problem lies in selecting the locations of base stations (BTSs) from a set of candidate sites, as well as their parameter settings, in such a way that a number of network requirements are satisfied. These requirements include the maximization of the covered area and the traffic capacity, while the infrastructure cost is to be minimized. Configuring BTSs is not a simple task, since it implies setting up many configuration parameters, such as the antenna type, the emission power, and/or the tilt and azimuth angles. However, cellular networks need to be adapted to the strongly competitive telecommunication industry: new services, new equipment technologies, increasing system capacity, etc. Even in relatively mature cellular markets, these issues force the deployment of additional sites not only to enhance the system capacity but also to provide increased levels of in-building coverage as mobile users expect to be offered service in all geographical area. The ACP problem therefore holds both for the second generation of cellular phone systems, GSM (*Global System for Mobile* communication, [7]), and its enhanced releases GPRS ([4]) and EDGE ([2]), as well as for the current third generation networks, UMTS (*Universal Mobile Telecommunication System*, [8]).

The ACP problem, as an extension of the classical minimum cost set covering problem, has NP-hard complexity ([3]). Also, the number of configurations for each BTS is very large, thus leading to a huge search space. Even when the BTS parameters such as emission power, tilt and azimuth, which are inherently continuous, are discretized by only considering a subset of possible values. An additional issue emerges in this optimization problem: changing the configuration setting of any BTS may affect other BTSs. For instance, if the maximum emission power of a BTS b is reduced to decrease the signal interference in a given area of the network, other BTSs should hold the traffic capacity that has been left unsupported by b. If these other BTSs are already operating at their full capacity, the network would simply start dropping calls of mobile users. This means that making small local changes would require to recompute most of the network predictions.

As the cellular networks become larger, the ACP problem requires a higher computation effort in order for it to be addressed in a reasonable amount of time. The aim of this deliverable is to analyze the performance of a Grid-enabled multiobjective EA, called *asynchronous steady state NSGA-II* (assNSGA-II), that allows the execution time of large instances to be reduced to affordable times.

2. Asynchronous steady state NSGA-II

The assNSGA-II algorithm is a steady state, Grid-enabled version of the NSGA-II algorithm that follows the master-slave paradigm, that is, the master executes the algorithm's main loop, and the slaves simply evaluate the tentative solutions. To deploy our algorithms in a distributed environment we have implemented a software tool called *sparrow*. Sparrow is inspired in MW [5], a C++ framework developed on top of Condor [10] to implement master-worker applications in grids.

assNSGA-II is based on NSGA-II [1], a generational algorithm, in which there is a current population that is used to create an auxiliary one (the offpring population); after that, both populations are combined to obtain the new current population. Typically, both the current and the auxiliary populations have the same size. An alternative to a generational GA is a steady state GA, which basically means that there is only a population, without using an auxiliar one. This way, new individuals are incorporated immediately in the evolutionary cycle, so parents and offsprings coexist in the same population. As a consequence, steady state GAs are characterized by incrementing the intensificacion capability of the search process. A steady state version of NSGA-II can be easily implemented by

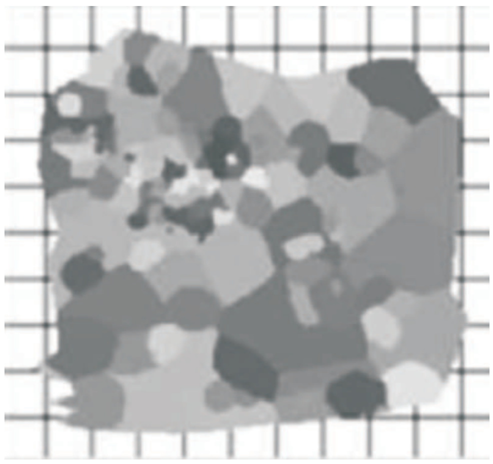

Figura 1: Topology of the instance Arno 3.1

Instance	Arno 3.1
Total traffic (Erlangs)	8 089.78
Number of sites	747
Number of test points ($\|\mathcal{ST}\|$)	42 975
Number of traffic test points ($\|\mathcal{T}\|$)	21 475

Table 1: Several values that characterize the instance tackled.

using an offspring population of size 1. This means that the ranking and crowding procedures have to be applied each time a new individual is created, so the computational complexity of the algorithm will increase notably.

In assNSGA-II, the master generates as many individuals as available processors. Once each evaluated solution is received, it is inserted in the population applying ranking and crowding. As in agNSGA-II, when idle workers are detected, new individuals are created and sent for evaluation.

3. Experimentation

3.1. ACP instance

The instance Arno 3.1 provided by France Telecom R&D has been used in this experimental study, which models a urban area. Its topology is displayed in Figure 1. Table 1 includes the most relevant values that characterize the instance.

3.2. Experimental Methodology

In order to measure the performance of the three multiobjective EAs used, the quality of their resulting nondominated set of solutions has to be considered. Since the exact Pareto fronts of the three ACP instances are unknown, a quality indicator which does not require this information beforehand is mandatory. This is why the *Hypervolume*, HV, indicator ([11]) has been chosen in this work. HV is one out of the most well-known and widely used indicator in the literature that allows two desirable features of Pareto fronts (convergence and diversity) to be measured with one single value. Convergence refers to the closeness of the approximated set towards Pareto optimal solutions whereas diversity considers the spread-out of this set of solutions along the nondominated front.

The hypervolume calculates the volume (in the objective space) covered by members of a non-dominated set of solutions Q (the region enclosed into the discontinuous line in Fig. 2, $Q = \{A, B, C\}$) for problems where all objectives are to be minimized. Mathematically, for each solution $i \in Q$, a hypercube v_i is constructed with a reference point W and the solution i as the diagonal corners of the hypercube. The reference point can simply be found by constructing a vector of worst objective function values. Thereafter, a union of all hypercubes is found and its hypervolume (HV) is calculated:

$$HV = volume\left(\bigcup_{i=1}^{|Q|} v_i\right).$$

Higher values of the hypervolume metrics are desirable. Since this indicator is not free from an arbitrary scaling of the objectives, the following normalization procedure has been performed in order to guarantee a fair comparison among the algorithms. First, all the nondominated solutions reached in all the executions of all the algorithms (for the same ACP instance) involved in the experimental study are collected into one single nondominated set.

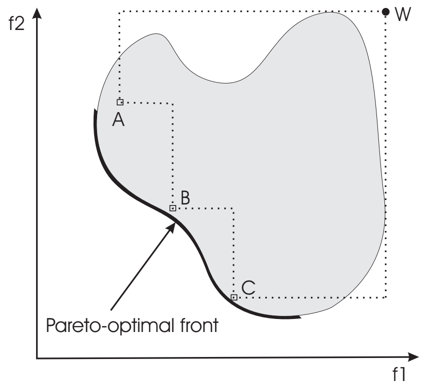

Figura 2: The hypervolume enclosed by the non-dominated solutions.

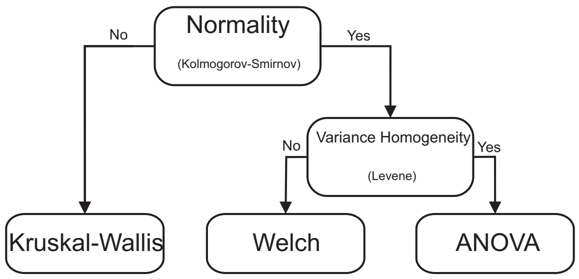

Figura 3: Statistical analysis performed in this work.

Dominated solutions are then removed. This resulting set can be considered the reference Pareto optimal set to be used as the basis for the normalization. By using the extreme values of this Pareto front, all the objectives of the approximated fronts reached by algorithms are mapped to [0.0,1.0] so that their scales have all the same influence in the HV indicator.

Since stochastic algorithms are considered and the results have to be provided with statistical significance, 30 independent runs for each algorithm and each problem instance have been done. The HV indicator is then computed for each of the approximated fronts (after the aforementioned normalization procedure), thus obtaining 30 HV values for each pair algorithm/instance. Next, the following statistical analysis has been carried out. First, a Kolmogorov-Smirnov test is performed in order to check whether the values of the results follow a normal (Gaussian) distribution or not. If so, the Levene test checks for the homogeneity of the variances. If samples have equal variance (positive Levene test), an ANOVA test is done; otherwise a Welch test is performed. For non-gaussian distributions, the non-parametric Kruskal-Wallis test is used to compare the medians of the algorithms. Figure 3 summarizes the statistical analysis. A confidence level of 95 % (i.e., significance level of 5 % or p-value under 0,05) is considered in the statistical tests. This means that the differences are unlikely to have occurred by chance with a probability of 95 %.

3.3. Results

The assNSGA-II algorithm has been compared to ssNSGA-II, the steady state version of NSGA-II which is the base algorithm of the Grid-enabled former. They two have used the same parameterization: a population size of 100 individuals, geographical crossover with $p_c = 0{,}9$, and multilevel mutation with $p_m = 1/L$ (being L the number of sites). The stopping condition has been to compute 25000 function evaluations. All the value in the tables are averaged over 30 independent runs.

Table 2: Performance of assNSGA-II for Arno 3.1.

Measure	Best value	$\bar{x}_{\pm\sigma_n}$
Number of slaves	288	$215_{\pm 41}$
Total CPU time (s)	410748 (4.75 days)	$505019_{\pm 26648}$ (5.85 days)
Wall-clock time (s)	2475 (0.69 hours)	$3799_{\pm 1463}$ (1.05 hours)
Parallel performance	95.64 %	$69{,}22\,\%_{\pm 16{,}10\,\%}$
ssNSGA-II time (s)	127836 (1.48 days)	$167577_{\pm 30578}$ (1.94 days)
Average parallel efficiency	20.52 %	

Table 3: HV values of ssNSGA-II and assNSGA-II for Arno 3.1.

	HV
Algorithm	$\bar{x}_{\pm\sigma_n}$
ssNSGA-II	$0{,}4766_{\pm 0{,}0095}$
assNSGA-II	$0{,}4801_{\pm 0{,}0081}$
Statistics	–

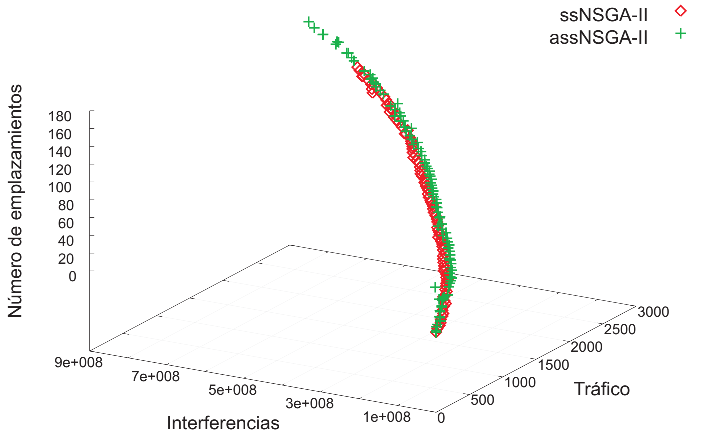

Figura 4: Approximated Pareto fronts of ssNSGA-II and assNSGA-II for Arno 3.1.

3.3.1. Parallel performance

The first study is devoted to analyzing the parallel performance of assNSGA-II. Table 2 includes the best values, the average values, $\bar{x}$, and the standard deviations, σ_n, of several performance measures. The first of these measures is the number of slaves involved in the parallel deployments of assNSGA-II. As it can be observed, 288 parallel processors have been used at most and 215 on average.

Table 2 also shows the total CPU time reported by Sparrow (i.e., the total accumulated CPU time used by all the processors involved in the parallel computation). It can be seen that this time is almost 6 days, whereas the actual computation wall-clock time is 1.05 hours (more than 132 times faster). This clearly points out the benefits of using a grid-enable metaheuristic.

If one considers the results of the parallel performance, Sparrow reports that this value is about 70 % on average, reaching 95 % in the best scenario. This is a remarkable behavior for an execution on a grid computing system. However, even though the evaluation time of the Arno 3.1 instance is long, sending all the network information for such evaluation through the network (about 1 MB) is also costly. This makes the computation/communication ratio to be somehow affected and thus the parallel efficiency is rather low.

The last two rows of Table 2 show both the ssNSGA-II computational time (sequential algorithm) and the actual parallel efficiency with respect to the assNSGA-II. This comparison makes the difference to drop down to 20 %. This can be explained because of the high communication overhead. However, in a practical setting, ssNSGA-II requires two days to compute the Pareto front of the Arno 3.1 instances, while assNSGA-II just needs one hour.

3.3.2. Numerical efficiency

This section analyzes the numerical efficiency of assNSGA-II with respect to its sequential counterpart, ssNSGA-II. The HV indicator has been used. Table 3 shows the average, $\bar{x}$, and the standard deviation, σ_n, of HV over 30 independent runs. The last row in the table also includes the results of the statistical analysis performed, which points out that no statistical differences have been found at 95 % of confidence level.

As it can be seen, assNSGA-II is able to approximate the front of Arno 3.1 with a better (higher) value of HV than ssNSGA-II. Despite the small, non significative differences, a quick glance at the resulting Pareto fronts (Figure 4) shows several advantages of the grid version over the sequential one. That is, both algorithms are able to converge towards the same region of the Pareto front (the approximations are practically overlapped), but assNSGA-II can reach regions for which ssNSGA-II find difficulties to cover (specially the region of a high number of sites, high traffic held and high interference).

The truly interesting fact heres is that the new algorithmic model engineered to profit from the grid computing platform is not only able to reduce the computational time (from days to hours), but also to better explore the search space. The key factor seems to be the asynchronism induced in assNSGA-II. That is, since the tentative solutions are sent out to the slaves for evaluation, the order in which they are returned back once evaluated is not the same and, therefore, a individual generated in the iteration t can be inserted back into the population in iteratioi $t + k$, $t \geq 0$,

$k > 0$. This induces a higher diversity in the assNSGA-II population which is translated into a better exploration of the search space.

Referencias

[1] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[2] A. Furuskar, J.Ñaslund, and H. Olofsson. EDGE – enhanced data rates for GSM and TDMA/136 evolution. *Ericsson Review*, (1), 1999.

[3] C. Glasser, S. Reith, and H. Vollmer. The complexity of base station positioning in cellular networks. *Discrete Applied Mathematics*, 148(1):1 – 12, 2005.

[4] H. Granbohm and J. Wiklund. GPRS – general packet radio service. *Ericsson Review*, (1), 1999.

[5] J. Linderoth, S. Kulkarni, J.P. Goux, and M. Yoder. An Enabling Framework for Master-Worker Applications on the Computational Grid. In *Proceedings of the Ninth IEEE Symposium on High Performance Distributed Computing (HPDC)*, pages 43–50, 2000.

[6] A. R. Mishra. *Fundamentals of Cellular Network Planning and Optimisation: 2G/2.5G/3G... Evolution to 4G.* Wiley, 2004.

[7] M. Mouly and M. B. Paulet. *The GSM System for Mobile Communications.* Mouly et Paulet, Palaiseau, 1992.

[8] J. Rapeli. UMTS: Targets, system concept, and standardization in a global framework. *IEEE Personal Communications*, 2(1):30 – 37, 1995.

[9] M. G. C. Resende and P. M. Pardalos, editors. *Handbook of Optimization in Telecommunications.* Springer, 2006.

[10] Douglas Thain, Todd Tannenbaum, and Miron Livny. Condor and the grid. In Fran Berman, Geoffrey Fox, and Tony Hey, editors, *Grid Computing: Making the Global Infrastructure a Reality.* John Wiley & Sons Inc., December 2002.

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