Apprentissage par transfert : du TSP au VRP

Ali Yaddaden[1], Sébastien Harispe[1], Michel Vasquez[1]

EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales, Ales, France
{prenom.nom}@mines-ales.fr

Mots-clés : *TSP, VRP, apprentissage profond, par renforcement, et par transfert*

1 Introduction

L'utilisation de l'apprentissage automatique pour la résolution de problèmes d'optimisation combinatoire attire l'attention des communautés de la RO et de l'apprentissage automatique. Un nouveau paradigme propose par exemple de créer des solveurs à base de réseaux de neurones profonds entraînés sur d'immenses bases de données d'instances à l'aide de l'apprentissage par renforcement. Ce paradigme a été testé sur le problème du voyageur de commerce (TSP) [5, 1] et les problèmes de tournées de véhicules avec contraintes de capacité (VRP) [3, 2] ; il obtient d'assez bons résultats pour des instances de petites tailles (< 100 clients).

Les modèles qui épousent ce paradigme sont aujourd'hui entraînés indépendemment pour le TSP et le VRP bien que les deux problèmes soient étroitement liés : un TSP est un VRP avec un seul tour, des clients avec une demande nulle, et une capacité infinie pour le véhicule. Dans ce contexte, nous conjecturons qu'un apprentissage par transfert de connaissances du domaine du TSP au domaine du VRP pourrait apporter une plus-value pour la résolution de ce dernier : réduire les temps d'entraînement et nombre d'instances requis. A notre connaissance aucune étude similaire n'a été faite dans le cadre du TSP et du VRP. Nos expériences préliminaires montrent qu'entraîner un réseau de neurones à résoudre le VRP en ayant préalablement été entraîné à résoudre le TSP offre un gain significatif en termes de vitesse d'apprentissage et de longueur de tours lorsque l'on a moins d'instances.

2 Apprentissage par transfert pour le VRP

Le cadre de l'apprentissage par transfert considère deux domaines appelés domaines source (D_s) et cible (D_c) et deux tâches T_s et T_c à traiter dans chacun des domaines [4]. Le but est alors d'exploiter les informations obtenues à partir de l'entraînement d'un réseau de neurones sur D_s pour la résolution de T_s afin d'apprendre à résoudre T_c sur D_c. Dans le cas d'un TSP 2D-Euclidien, D_s correspond à l'espace des caractéristiques d'un TSP, à savoir, les coordonnées des villes à visiter tandis que T_s correspond à la tournée. D_c est l'espace des caractéristiques d'un VRP (les coordonnées des clients et du dépôt, les demandes des clients et la capacité du véhicule) et T_c est l'ordre de parcours des clients suivant les contraintes des véhicules.

Il s'agit alors, dans les deux cas, de trouver une politique paramétrée P_θ capable pour une instance de n villes (clients + dépôt dans le cas du VRP) $X = \{x_0, ..., x_{n-1}\}$ de définir l'ordre de parcours $Y = (y_0, ..., y_{K-1})$ ($K = n$ pour le TSP et $K \geq n$ pour le VRP). Cette politique est donnée par la formule :

$$P_\theta(Y|X) = \prod_{i=0}^{K-1} p_\theta(y_i|y_0, ..., y_{i-1}, X) \tag{1}$$

Les paramètres θ de la politique sont généralement identifiés grâce à un algorithme d'apprentissage par renforcement [1] avec comme fonction de récompense la longueur du tour (TSP) ou des tournées (VRP).

Pour nos expérimentations, nous avons utilisé un modèle issu de l'état de l'art à base de réseaux de neurones profonds capables de représenter la structure de graphe complet de nos instances [2]. Nous y avons apporté des modifications pour avoir un modèle avec le même nombre de paramètres pour la résolution du TSP et du VRP. Les entraînements ont été effectués avec un nombre variable d'instances par itération d'apprentissage (époque) afin d'étudier l'apport de l'apprentissage par transfert suivant ce facteur. Nous avons au préalable entraîné notre modèle à résoudre le TSP selon les modalités mentionnées dans [2].Le même modèle a de nouveau été utilisé pour apprendre à résoudre le VRP en diminuant la vitesse d'apprentissage. Pour comparer, nous avons entraîné un modèle à résoudre le VRP sans apprentissage par transfert avec le mêmes nombre d'instances que le modèle entraîné par apprentissage par transfert.

Nous avons effectué des tests préliminaires sur des instances de VRP à 20 clients [1] en considérants des nombres variables d'instances par itération : 16K, 32K et 64K. La figure 1 illustre l'évolution de la moyenne des longueurs des tours des instances par itération lors de la phase d'entraînement. Nos résultats préliminaires sont encourageants : dans notre contexte de test, le transfert (i) accélère l'apprentissage de la résolution du VRP et (ii) améliore la longueur des tours trouvés.

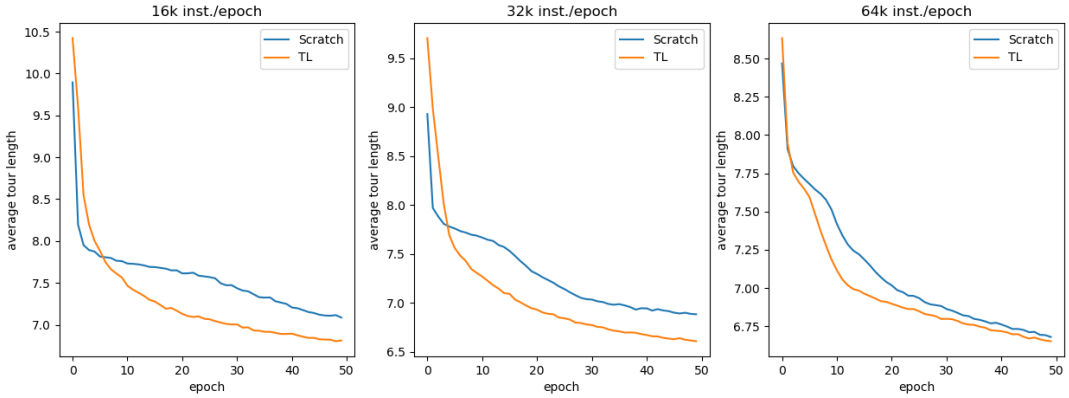

FIG. 1 – Longueurs moyennes des tours par itération sur le jeu d'entraînement pour un modèle VRP20 entraîné de zéro (Scratch/bleu) et entraîné par apprentissage par transfert (TL/orange).

Références

[1] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv :1611.09940*, 2016.

[2] Wouter Kool, Herke Van Hoof, and Max Welling. Attention, learn to solve routing problems ! *arXiv preprint arXiv :1803.08475*, 2018.

[3] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence V Snyder, and Martin Takáč. Reinforcement learning for solving the vehicle routing problem. *arXiv preprint arXiv :1802.04240*, 2018.

[4] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10) :1345–1359, 2009.

[5] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. *arXiv preprint arXiv :1506.03134*, 2015.

1. les coordonnées en 2 dimensions sont générées uniformément sur le carré $[0, 1]^2$. Les demandes des clients sont générées uniformément dans l'intervalle $[1, 9]$ et la capacité des véhicules est de 30.